# Table of Contents

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>What Is Amazon SES?</td>
<td>1</td>
</tr>
<tr>
<td>Why use Amazon SES?</td>
<td>1</td>
</tr>
<tr>
<td>Amazon SES and other AWS services</td>
<td>1</td>
</tr>
<tr>
<td>In this guide</td>
<td>2</td>
</tr>
<tr>
<td>Quick Start</td>
<td>3</td>
</tr>
<tr>
<td>Step 1: Sign up for AWS</td>
<td>3</td>
</tr>
<tr>
<td>Step 2: Verify your email address</td>
<td>3</td>
</tr>
<tr>
<td>Step 3: Send your first email</td>
<td>3</td>
</tr>
<tr>
<td>Step 4: Consider how you will handle bounces and complaints</td>
<td>3</td>
</tr>
<tr>
<td>Step 5: Move out of the Amazon SES sandbox</td>
<td>3</td>
</tr>
<tr>
<td>Next steps</td>
<td>4</td>
</tr>
<tr>
<td>Sending Email</td>
<td>5</td>
</tr>
<tr>
<td>How do I send emails using Amazon SES?</td>
<td>5</td>
</tr>
<tr>
<td>How do I start?</td>
<td>6</td>
</tr>
<tr>
<td>Concepts</td>
<td>7</td>
</tr>
<tr>
<td>Amazon SES and Deliverability</td>
<td>7</td>
</tr>
<tr>
<td>Email-Sending Process</td>
<td>11</td>
</tr>
<tr>
<td>Email Format and Amazon SES</td>
<td>14</td>
</tr>
<tr>
<td>Getting Started</td>
<td>16</td>
</tr>
<tr>
<td>Using the Amazon SES Console</td>
<td>17</td>
</tr>
<tr>
<td>Using Simple Mail Transfer Protocol (SMTP)</td>
<td>17</td>
</tr>
<tr>
<td>Using an AWS SDK</td>
<td>17</td>
</tr>
<tr>
<td>Before You Begin</td>
<td>17</td>
</tr>
<tr>
<td>Send an Email Using the Console</td>
<td>18</td>
</tr>
<tr>
<td>Send an Email Using SMTP</td>
<td>19</td>
</tr>
<tr>
<td>Send an Email Using an AWS SDK</td>
<td>28</td>
</tr>
<tr>
<td>Setting up Email</td>
<td>42</td>
</tr>
<tr>
<td>Signing up for AWS</td>
<td>42</td>
</tr>
<tr>
<td>Verifying Identities</td>
<td>43</td>
</tr>
<tr>
<td>Getting Your AWS Access Keys</td>
<td>57</td>
</tr>
<tr>
<td>Downloading an AWS SDK</td>
<td>57</td>
</tr>
<tr>
<td>Using a Custom MAIL FROM Domain</td>
<td>57</td>
</tr>
<tr>
<td>Setting up SPF Records</td>
<td>63</td>
</tr>
<tr>
<td>Getting Your SMTP Credentials</td>
<td>64</td>
</tr>
<tr>
<td>Moving Out of the Sandbox</td>
<td>64</td>
</tr>
<tr>
<td>Sending Your Email</td>
<td>65</td>
</tr>
<tr>
<td>Using the SMTP Interface</td>
<td>66</td>
</tr>
<tr>
<td>Using the API</td>
<td>99</td>
</tr>
<tr>
<td>Sending Personalized Email</td>
<td>107</td>
</tr>
<tr>
<td>Authenticating Your Email</td>
<td>115</td>
</tr>
<tr>
<td>Authenticating Email with SPF</td>
<td>116</td>
</tr>
<tr>
<td>Authenticating Email with DKIM</td>
<td>117</td>
</tr>
<tr>
<td>Complying with DMARC</td>
<td>126</td>
</tr>
<tr>
<td>Managing Your Sending Limits</td>
<td>127</td>
</tr>
<tr>
<td>Monitoring Your Sending Limits</td>
<td>128</td>
</tr>
<tr>
<td>Increasing Your Sending Limits</td>
<td>129</td>
</tr>
<tr>
<td>What Happens When You Reach Your Sending Limits</td>
<td>131</td>
</tr>
<tr>
<td>Using Sending Authorization</td>
<td>132</td>
</tr>
<tr>
<td>Overview of Sending Authorization</td>
<td>132</td>
</tr>
<tr>
<td>Sending Authorization Policies</td>
<td>134</td>
</tr>
<tr>
<td>Sending Authorization Policy Examples</td>
<td>138</td>
</tr>
<tr>
<td>Identity Owner Tasks</td>
<td>144</td>
</tr>
<tr>
<td>Delegate Sender Tasks</td>
<td>149</td>
</tr>
<tr>
<td>Using Dedicated IP Addresses</td>
<td>156</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Common Domain Verification Problems</td>
<td>425</td>
</tr>
<tr>
<td>How to Check Domain Verification Settings</td>
<td>426</td>
</tr>
<tr>
<td>DKIM Problems</td>
<td>427</td>
</tr>
<tr>
<td>Notification Problems</td>
<td>428</td>
</tr>
<tr>
<td>Removing an Email Address from the Suppression List</td>
<td>428</td>
</tr>
<tr>
<td>Increasing Throughput</td>
<td>429</td>
</tr>
<tr>
<td>SMTP Issues</td>
<td>430</td>
</tr>
<tr>
<td>SMTP Response Codes</td>
<td>433</td>
</tr>
<tr>
<td>FAQs</td>
<td>436</td>
</tr>
<tr>
<td>Enforcement FAQs</td>
<td>436</td>
</tr>
<tr>
<td>Probations</td>
<td>436</td>
</tr>
<tr>
<td>Suspensions</td>
<td>438</td>
</tr>
<tr>
<td>Bounces</td>
<td>440</td>
</tr>
<tr>
<td>Complaints</td>
<td>442</td>
</tr>
<tr>
<td>Spamtraps</td>
<td>446</td>
</tr>
<tr>
<td>Manual Investigations</td>
<td>448</td>
</tr>
<tr>
<td>IP Blacklist FAQs</td>
<td>449</td>
</tr>
<tr>
<td>Blacklist FAQ Q1</td>
<td>449</td>
</tr>
<tr>
<td>Blacklist FAQ Q2</td>
<td>449</td>
</tr>
<tr>
<td>Blacklist FAQ Q3</td>
<td>450</td>
</tr>
<tr>
<td>Blacklist FAQ Q4</td>
<td>450</td>
</tr>
<tr>
<td>Blacklist FAQ Q5</td>
<td>450</td>
</tr>
<tr>
<td>Email Metrics FAQs</td>
<td>450</td>
</tr>
<tr>
<td>General</td>
<td>451</td>
</tr>
<tr>
<td>Open Tracking</td>
<td>451</td>
</tr>
<tr>
<td>Click Tracking</td>
<td>452</td>
</tr>
<tr>
<td>Resources</td>
<td>454</td>
</tr>
<tr>
<td>Appendix</td>
<td>455</td>
</tr>
<tr>
<td>Appendix: Header Fields</td>
<td>455</td>
</tr>
<tr>
<td>Appendix: Unsupported Attachment Types</td>
<td>456</td>
</tr>
<tr>
<td>Appendix: Create a Deliverability Dashboard</td>
<td>457</td>
</tr>
<tr>
<td>Part 1: Create an Amazon SNS Topic</td>
<td>459</td>
</tr>
<tr>
<td>Part 2: Create an Amazon SQS Queue</td>
<td>459</td>
</tr>
<tr>
<td>Part 3: Create an Amazon S3 Bucket</td>
<td>460</td>
</tr>
<tr>
<td>Part 4: Create IAM Policy and Role</td>
<td>460</td>
</tr>
<tr>
<td>Part 5: Configure Amazon SES Notifications</td>
<td>462</td>
</tr>
<tr>
<td>Part 6: Create a Lambda Function</td>
<td>462</td>
</tr>
<tr>
<td>Part 7: Test the Lambda Function</td>
<td>464</td>
</tr>
<tr>
<td>Part 8: Configure CloudWatch Triggers</td>
<td>464</td>
</tr>
<tr>
<td>Document History</td>
<td>465</td>
</tr>
</tbody>
</table>
What Is Amazon SES?

Welcome to the Amazon Simple Email Service (Amazon SES) Developer Guide. Amazon SES is an email platform that provides an easy, cost-effective way for you to send and receive email using your own email addresses and domains.

For example, you can send marketing emails such as special offers, transactional emails such as order confirmations, and other types of correspondence such as newsletters. When you use Amazon SES to receive mail, you can develop software solutions such as email autoresponders, email unsubscribe systems, and applications that generate customer support tickets from incoming emails.

You only pay for what you use, so you can send and receive as much or as little email as you like. For service highlights, FAQs, and pricing information, go to the Amazon SES Detail Page.

Why use Amazon SES?

Building a large-scale email solution is often a complex and costly challenge for a business. You must deal with infrastructure challenges such as email server management, network configuration, and IP address reputation. Additionally, many third-party email solutions require contract and price negotiations, as well as significant up-front costs. Amazon SES eliminates these challenges and enables you to benefit from the years of experience and sophisticated email infrastructure Amazon.com has built to serve its own large-scale customer base.

Amazon SES and other AWS services

Amazon SES integrates seamlessly with other AWS products. For example, you can:

- Add email capabilities to any application that runs on an Amazon Elastic Compute Cloud (Amazon EC2) instance by using the AWS SDKs or the Amazon SES API. If you want to send email through Amazon SES from an Amazon EC2 instance, you can get started with Amazon SES for free.
- Use AWS Elastic Beanstalk to create an email-enabled application such as a program that uses Amazon SES to send a newsletter to customers.
- Set up Amazon Simple Notification Service (Amazon SNS) to notify you of your emails that bounced, produced a complaint, or were successfully delivered to the recipient's mail server. When you use Amazon SES to receive emails, your email content can be published to Amazon SNS topics.
- Use the AWS Management Console to set up Easy DKIM, which is a way to authenticate your emails. Although you can use Easy DKIM with any DNS provider, it is especially easy to set up when you manage your domain with Route 53.
- Control user access to your email sending by using AWS Identity and Access Management (IAM).
- Store emails you receive in Amazon Simple Storage Service (Amazon S3).
- Take action on your received emails by triggering AWS Lambda functions.
- Use AWS Key Management Service (AWS KMS) to optionally encrypt the mail you receive in your Amazon S3 bucket.
- Use AWS CloudTrail to log Amazon SES API calls that you make using the console or the Amazon SES API.
- Publish your email sending events to Amazon CloudWatch or Amazon Kinesis Firehose. If you publish your email sending events to Kinesis Firehose, you can access them in Amazon Redshift, Amazon Elasticsearch Service, or Amazon S3.
In this guide

This guide contains the following sections:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sending Email (p. 5)</td>
<td>Describes how you can send email using Amazon SES.</td>
</tr>
<tr>
<td>Receiving Email (p. 166)</td>
<td>Describes how you can receive email using Amazon SES.</td>
</tr>
<tr>
<td>Controlling Access (p. 354)</td>
<td>Shows you how to use Amazon SES with AWS Identity and Access Management (IAM) to specify which Amazon SES API actions a user can perform on which Amazon SES resources.</td>
</tr>
<tr>
<td>Logging API Calls (p. 360)</td>
<td>Provides a list of Amazon SES APIs that can be logged using AWS CloudTrail.</td>
</tr>
<tr>
<td>Using Credentials (p. 369)</td>
<td>Explains the types of credentials that you might use with Amazon SES, and when you might use them.</td>
</tr>
<tr>
<td>Using the API (p. 371)</td>
<td>Describes how to use the Amazon SES Query API.</td>
</tr>
<tr>
<td>Regions (p. 409)</td>
<td>Lists the Amazon SES SMTP and API endpoints for the AWS regions in which Amazon SES is available, and contains information you need to know when you use Amazon SES endpoints in multiple regions.</td>
</tr>
<tr>
<td>Limits (p. 413)</td>
<td>Provides a list of limits within Amazon SES.</td>
</tr>
<tr>
<td>Resources (p. 454)</td>
<td>Lists resources that you may find useful as you work with Amazon SES.</td>
</tr>
<tr>
<td>Appendix (p. 455)</td>
<td>Provides supplementary information about header fields, unsupported attachment types, and scripts.</td>
</tr>
</tbody>
</table>
Amazon SES Quick Start

This procedure leads you through the steps to sign up for AWS, verify your email address, send your first email, consider how you will handle bounces and complaints, and move out of the Amazon Simple Email Service (Amazon SES) sandbox.

Use this procedure if you:

- Are just experimenting with Amazon SES.
- Want to send some test emails without doing any programming.
- Want to get set up in as few steps as possible.

Step 1: Sign up for AWS

Before you can use Amazon SES, you need to sign up for AWS. When you sign up for AWS, your account is automatically signed up for all AWS services.

For instructions, see Signing up for AWS (p. 42).

Step 2: Verify your email address

Before you can send email from your email address through Amazon SES, you need to show Amazon SES that you own the email address by verifying it.

For instructions, see Verifying Email Addresses in Amazon SES (p. 43).

Step 3: Send your first email

You can send an email simply by using the Amazon SES console. As a new user, your account is in a test environment called the sandbox, so you can only send email to and from email addresses that you have verified.

For instructions, see Send an Email Using the Amazon SES Console (p. 18).

Step 4: Consider how you will handle bounces and complaints

Before the next step, you need to think about how you will handle bounces and complaints. If you are sending to a small number of recipients, your process can be as simple as examining the bounce and complaint feedback that you receive by email, and then removing those recipients from your mailing list.

Step 5: Move out of the Amazon SES sandbox

To be able to send emails to unverified email addresses and to raise the number of emails you can send per day and how fast you can send them, your account needs to be moved out of the sandbox. This process involves opening an SES Sending Limits Increase case in Support Center.
For more information about the sandbox restrictions and how to apply to move out of the sandbox, see Moving Out of the Amazon SES Sandbox (p. 64).

Next steps

- After you send a few test emails to yourself, use the Amazon SES mailbox simulator for further testing because emails to the mailbox simulator do not count towards your sending quota or your bounce and complaint rates. For more information on the mailbox simulator, see Testing Amazon SES Email Sending (p. 162).
- Monitor your sending activity, such as the number of emails that you have sent and the number that have bounced or received complaints. For more information, see Monitoring Your Amazon SES Sending Activity (p. 221).
- Verify entire domains so that you can send email from any email address in your domain without verifying addresses individually. For more information, see Verifying Domains in Amazon SES (p. 52).
- Increase the chance that your emails will be delivered to your recipients' inboxes instead of junk boxes by authenticating your emails. For more information, see Authenticating Your Email in Amazon SES (p. 115).
Sending Email with Amazon SES

When you send an email, you are sending it through some type of outbound email server. That email server might be provided by your Internet service provider (ISP), your company's IT department, or you might have set it up yourself. The email server accepts your email content, formats it to comply with email standards, and then sends the email out over the Internet. The email may pass through other servers until it eventually reaches a receiver (an entity, such as an ISP, that receives the email on behalf of the recipient). The receiver then delivers the email to the recipient. The following diagram illustrates the basic email-sending process.

When you use Amazon SES, Amazon SES becomes your outbound email server. You can also keep your existing email server and configure it to send your outgoing emails through Amazon SES so that you don't have to change any settings in your email clients. The following diagram shows where Amazon SES fits in to the email sending process.

A sender can generate the email content in different ways. A sender can create the email by using an email client application, or use a program that automatically generates emails, like an application that sends order confirmations in response to purchase transactions.

How do I send emails using Amazon SES?

There are several ways that you can send an email by using Amazon SES. You can use the Amazon SES console, the Simple Mail Transfer Protocol (SMTP) interface, or you can call the Amazon SES API.

- **Amazon SES console**—This method is the quickest way to set up your system and send a couple of test emails, but once you are ready to start your email campaign, you will use the console primarily to monitor your sending activity. For example, you can quickly view the number of emails that you have sent and the number of bounces and complaints that you have received.

- **SMTP Interface**—There are two ways to access Amazon SES through the SMTP interface. The first way, which requires no coding, is to configure any SMTP-enabled software to send email through Amazon SES. For example, you can configure your existing email client or software program to connect to the Amazon SES SMTP endpoint instead of your current outbound email server.

   The second way is to use an SMTP-compatible programming language such as Java and access the Amazon SES SMTP interface by using the language's built-in SMTP functions and data types.

- **Amazon SES API**—You can call the Amazon SES Query API directly through HTTPS, or you can use the AWS Command Line Interface, the AWS Tools for Windows PowerShell, or an AWS SDK. The AWS SDKs wrap the low-level functionality of the Amazon SES API with higher-level data types and function calls that take care of the details for you. The AWS SDKs provide not only Amazon SES operations, but also basic AWS functionality such as request authentication, request retries, and error handling.
How do I start?

If you are a first-time user of Amazon SES, we recommend that you begin by reading the following sections:

- **Amazon SES Quick Start (p. 3)**—Shows you how to get set up and send a test email as quickly as possible.
- **Getting Started Sending Email with Amazon SES (p. 16)**—Shows you how to send an email by using the Amazon SES console, the SMTP interface, and an AWS SDK. Examples are provided in C#, Java, and PHP.
- **Amazon SES and Deliverability (p. 7)**—Explains email deliverability concepts that you should be familiar with when you use Amazon SES.
- **Amazon SES Email-Sending Process (p. 11)**—Shows you what happens when you send an email through Amazon SES.
- **Email Format and Amazon SES (p. 14)**—Reviews the format of emails and identifies the information that you need to provide to Amazon SES.

Then you can learn about sending email with Amazon SES in more detail by reading the sections listed in the following table:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting up Email (p. 42)</td>
<td>Shows you how to sign up for AWS, get your AWS access keys, download an AWS SDK, verify email addresses or domains, and move out of the Amazon SES sandbox.</td>
</tr>
<tr>
<td>Using the SMTP Interface (p. 66)</td>
<td>Shows you how to get your Amazon SES SMTP credentials, connect to the Amazon SES SMTP endpoint, and provides examples of how to configure email clients and software packages to send email through Amazon SES. Also explains how to configure your existing email server to send all outgoing emails through Amazon SES.</td>
</tr>
<tr>
<td>Using the API (p. 99)</td>
<td>Shows you how to send formatted and raw emails by using the Amazon SES API. Explains how to use non-standard characters and send attachments by using the Multipurpose Internet Mail Extensions (MIME) standard when you send raw emails.</td>
</tr>
<tr>
<td>Authenticating Your Email (p. 115)</td>
<td>Shows you how to use DKIM with Amazon SES to show ISPs that you own the domain you are sending from.</td>
</tr>
<tr>
<td>Managing Your Sending Limits (p. 127)</td>
<td>Explains the two Amazon SES sending limits (sending quota and maximum send rate), how to increase them, and the errors you receive when you try to exceed them.</td>
</tr>
<tr>
<td>Using Sending Authorization (p. 132)</td>
<td>Shows you how to authorize other users to send emails from your identities on your behalf.</td>
</tr>
<tr>
<td>Using Dedicated IP Addresses (p. 156)</td>
<td>You decide whether to use shared IP addresses or lease dedicated IP addresses for your Amazon SES sending. Provides procedures for requesting and relinquishing dedicated IPs, and for creating pools of dedicated IPs.</td>
</tr>
<tr>
<td>Testing Email Sending (p. 162)</td>
<td>Explains how to use the Amazon SES mailbox simulator to simulate common email scenarios without affecting</td>
</tr>
</tbody>
</table>
Amazon SES Email-Sending Concepts

The following sections contain fundamental information about how Amazon SES sends your mail.

- Amazon SES and Deliverability (p. 7)
- Amazon SES Email-Sending Process (p. 11)
- Email Format and Amazon SES (p. 14)

Amazon SES and Deliverability

You want your recipients to read your emails, find them valuable, and not label them as spam. In other words, you want to maximize email deliverability—the percentage of your emails that arrive in your recipients' inboxes. This topic reviews email deliverability concepts that you should be familiar with when you use Amazon SES.

To maximize email deliverability, you need to understand email delivery issues, proactively take steps to prevent them, stay informed of the status of the emails that you send, and then improve your email-sending program, if necessary, to further increase the likelihood of successful deliveries. The following sections review the concepts behind these steps and how Amazon SES helps you through the process.
Understand Email Delivery Issues

In most cases, your messages are delivered successfully to recipients who expect them. In some cases, however, a delivery might fail, or a recipient might not want to receive the mail that you are sending. Bounces, complaints, and the suppression list are related to these delivery issues and are described in the following sections.

Bounce

If your recipient's receiver (for example, an ISP) fails to deliver your message to the recipient, the receiver bounces the message back to Amazon SES. Amazon SES then notifies you of the bounced email through email or through Amazon Simple Notification Service (Amazon SNS), depending on how you have your system set up. For more information, see Monitoring Using Amazon SES Notifications (p. 226).

There are hard bounces and soft bounces, as follows:

- **Hard bounce** – A persistent email delivery failure. For example, the mailbox does not exist. Amazon SES does not retry hard bounces, with the exception of DNS lookup failures. We strongly recommend that you do not make repeated delivery attempts to email addresses that hard bounce.

- **Soft bounce** – A temporary email delivery failure. For example, the mailbox is full, there are too many connections (also called throttling), or the connection times out. Amazon SES retries soft bounces multiple times. If the email still cannot be delivered, then Amazon SES stops retrying it.
Amazon SES notifies you of hard bounces and soft bounces that will no longer be retried. However, only hard bounces count toward your bounce rate and the bounce metric that you retrieve using the Amazon SES console or the GetSendStatistics API.

Bounces can also be synchronous or asynchronous. A synchronous bounce occurs while the email servers of the sender and receiver are actively communicating. An asynchronous bounce occurs when a receiver initially accepts an email message for delivery and then subsequently fails to deliver it to the recipient.

**Complaint**

Most email client programs provide a button labeled "Mark as Spam," or similar, which moves the message to a spam folder, and forwards it to the ISP. Additionally, most ISPs maintain an abuse address (e.g., abuse@example.net), where users can forward unwanted email messages and request that the ISP take action to prevent them. In both of these cases, the recipient is making a complaint. If the ISP concludes that you are a spammer, and Amazon SES has a feedback loop set up with the ISP, then the ISP will send the complaint back to Amazon SES. When Amazon SES receives such a complaint, it forwards the complaint to you either by email or by using an Amazon SNS notification, depending on how you have your system set up. For more information, see Monitoring Using Amazon SES Notifications (p. 226). We recommend that you do not make repeated delivery attempts to email addresses that generate complaints.

**Suppression List**

The Amazon SES suppression list is a list of recipient email addresses that have recently caused a hard bounce for any Amazon SES customer. If you try to send an email through Amazon SES to an address that is on the suppression list, the call to Amazon SES succeeds, but Amazon SES treats the email as a hard bounce instead of attempting to send it. Like any hard bounce, suppression list bounces count towards your sending quota and your bounce rate. An email address can remain on the suppression list for up to 14 days. If you are sure that the email address that you're trying to send to is valid, you can submit a suppression list removal request. For more information, see Removing an Email Address from the Amazon SES Suppression List (p. 428).

**Be Proactive**

One of the biggest issues with email on the Internet is unsolicited bulk email, or spam. ISPs take considerable measures to prevent their customers from receiving spam. Correspondingly, Amazon SES takes proactive steps to decrease the likelihood that ISPs consider your email to be spam. Amazon SES uses verification, authentication, sending limits, and content filtering. Amazon SES also maintains a trusted reputation with ISPs and requires you to send high-quality email. Amazon SES does some of those things for you automatically (like content filtering); in other cases, it provides the tools (like authentication), or guides you in the right direction (sending limits). The following sections provide more information about each concept.

**Verification**

Unfortunately, it's possible for a spammer to falsify an email header and spoof the originating email address so that it appears as though the email originated from a different source. To maintain trust between ISPs and Amazon SES, Amazon SES needs to ensure that its senders are who they say they are. You are therefore required to verify all email addresses from which you send emails through Amazon SES to protect your sending identity. You can verify email addresses by using the Amazon SES console or by using the Amazon SES API. You can also verify entire domains. For more information, see Verifying Email Addresses in Amazon SES (p. 43) and Verifying Domains in Amazon SES (p. 52).

If your account is still in the Amazon SES sandbox, you also need to verify all recipient addresses except for addresses provided by the Amazon SES mailbox simulator. For information about getting out of the sandbox, see Moving Out of the Amazon SES Sandbox (p. 64). For more information about the mailbox simulator, see Testing Amazon SES Email Sending (p. 162).
Authentication

Authentication is another way that you can indicate to ISPs that you are who you say you are. When you authenticate an email, you provide evidence that you are the owner of the account and that your emails have not been modified in transit. In some cases, ISPs refuse to forward email that is not authenticated. Amazon SES supports two methods of authentication: Sender Policy Framework (SPF) and DomainKeys Identified Mail (DKIM). For more information, see Authenticating Your Email in Amazon SES (p. 115).

Sending Limits

If an ISP detects sudden, unexpected spikes in the volume or rate of your emails, the ISP might suspect you are a spammer and block your emails. Therefore, every Amazon SES account has a set of sending limits to regulate the number of email messages that you can send and the rate at which you can send them. These sending limits help you to gradually ramp up your sending activity to protect your trustworthiness with ISPs.

Amazon SES has two sending limits: a sending quota (the maximum number of messages you can send in a 24-hour period) and a maximum send rate (the maximum number of emails that Amazon SES can accept from your account per second, although the actual rate at which Amazon SES accepts your messages might be less than the maximum send rate). If you are a brand-new user, Amazon SES lets you send a small amount of email each day. If the mail that you send is acceptable to ISPs, this limit will gradually increase. Over time, your sending limits will steadily increase so that you can send larger quantities of email at faster rates. You can also file an SES Sending Limits Increase case to get your quotas increased if you need them to ramp up more quickly.

For more information about sending limits and how to increase them, see Managing Your Amazon SES Sending Limits (p. 127).

Content Filtering

Many ISPs use content filtering to determine if incoming emails are spam. Content filters look for questionable content and block the email if the email fits the profile of spam. Amazon SES uses content filters also. When your application sends a request to Amazon SES, Amazon SES assembles an email message on your behalf and then scans the message header and body to determine if they contain content that ISPs might construe as spam. If your messages look like spam to the content filters that Amazon SES uses, your reputation with Amazon SES will be negatively affected. If a message is infected with a virus, it is rejected by Amazon SES entirely.

Reputation

When it comes to email sending, reputation—a measure of confidence that an IP address, email address, or sending domain is not the source of spam—is important. Amazon SES maintains a strong reputation with ISPs so that ISPs deliver your emails to your recipients' inboxes. Similarly, you need to maintain a trusted reputation with Amazon SES. You build your reputation with Amazon SES by sending high-quality content. When you send high-quality content, your reputation becomes more trusted over time and Amazon SES increases your sending limits. Excessive bounces and complaints negatively impact your reputation and can cause Amazon SES to lower your sending limits or terminate your Amazon SES account.

One way to help maintain your reputation is to use the mailbox simulator when you test your system, instead of sending to email addresses that you have created yourself. Emails to the mailbox simulator do not count toward your bounce and complaint metrics. For more information about the mailbox simulator, see Testing Amazon SES Email Sending (p. 162).

High-Quality Email

High-quality email is email that recipients find valuable and want to receive. Value means different things to different recipients and can come in the form of offers, order confirmations, receipts,
newsletters, etc. Ultimately, your deliverability rests on the quality of the emails that you send because ISPs block emails that they find to be low quality (spam).

Stay Informed

Whether your deliveries fail, your recipients complain about your emails, or Amazon SES successfully delivers an email to a recipient's mail server, Amazon SES helps you to track down the issue by providing notifications and by enabling you to easily monitor your usage statistics.

Notifications

When an email bounces, the ISP notifies Amazon SES, and Amazon SES notifies you. Amazon SES notifies you of hard bounces and soft bounces that Amazon SES will no longer retry. Many ISPs also forward complaints, and Amazon SES sets up complaint feedback loops with the major ISPs so you don't have to. Amazon SES can notify you of bounces, complaints, and successful deliveries in two ways: you can set your account up to receive notifications through Amazon SNS, or you can receive notifications by email (bounces and complaints only). For more information, see Monitoring Using Amazon SES Notifications (p. 226).

Usage Statistics

Amazon SES provides usage statistics so that you can view your failed deliveries to determine and resolve the root causes. You can view your usage statistics by using the Amazon SES console or by calling the Amazon SES API. You can view how many deliveries, bounces, complaints, and virus-infected rejected emails you have, and you can also view your sending limits to ensure that you stay within them.

Improve Your Email-Sending Program

If you are getting large numbers of bounces and complaints, it's time to reassess your email-sending strategy. Remember that excessive bounces, complaints, and attempts to send low-quality email constitute abuse and put your AWS account at risk of termination. Ultimately, you need to be sure that you use Amazon SES to send high-quality emails and to only send emails to recipients who want to receive them.

Amazon SES Email-Sending Process

This topic describes what happens when you send an email with Amazon SES, and the various outcomes that can occur after the email is sent. The following figure is a high-level overview of the sending process:

1. A client application, acting as an email sender, makes a request to Amazon SES to send email to one or more recipients.
2. If the request is valid, Amazon SES accepts the email and sends it over the Internet to the recipient's receiver. Once the message is passed to Amazon SES, it is usually sent immediately, with the first delivery attempt normally occurring within milliseconds.

3. At this point, there are different possibilities. For example:
   a. The ISP successfully delivers the message to the recipient's inbox.
   b. The recipient's email address does not exist, so the ISP sends a bounce notification to Amazon SES. Amazon SES then forwards the notification to the sender.
   c. The recipient receives the message but considers it to be spam and registers a complaint with the ISP. The ISP, which has a feedback loop set up with Amazon SES, sends the complaint to Amazon SES, which then forwards it to the sender.

The following sections review the individual possible outcomes after a sender sends an email request to Amazon SES and after Amazon SES sends an email message to the recipient.

### After a Sender Sends an Email Request to Amazon SES

When the sender makes a request to Amazon SES to send an email, the call may succeed or fail. The following sections describe what happens in each case.

#### Successful Sending Request

If the request to Amazon SES succeeds, Amazon SES returns a success response to the sender. This message includes the **message ID**, a string of characters that uniquely identifies the request. You can use the message ID to identify the sent email or to track problems encountered during sending. Amazon SES then assembles an email message based on the request parameters, scans the message for questionable content and viruses and then sends it out over the Internet using Simple Mail Transfer Protocol (SMTP). Your message is usually sent immediately; the first delivery attempt typically occurs within milliseconds.

**Note**

If Amazon SES successfully accepts the sender's request and then an Amazon SES content filter finds that the message contains a virus, Amazon SES drops the message and notifies the sender by email.

#### Failed Sending Request

If the sender's email-sending request to Amazon SES fails, Amazon SES responds to the sender with an error and drops the email. The request could fail for several reasons. For example, the request may not be formatted properly or the email address may not have been verified by the sender.

The method through which you can determine if the request has failed depends on how you call Amazon SES. The following are examples of how errors and exceptions are returned:

- If you are calling Amazon SES through the Query (HTTPS) API (SendEmail or SendRawEmail), the actions will return an error. For more information, see the Amazon Simple Email Service API Reference.
- If you are using an AWS SDK for a programming language that uses exceptions, the call to Amazon SES will throw a *MessageRejectedException*. (The name of the exception may vary slightly depending on the SDK.)
- If you are using the SMTP interface, then the sender receives an SMTP response code, but how the error is conveyed depends on the sender's client. Some clients may display an error code; others may not.

For information about errors that can occur when you send an email with Amazon SES, see Amazon SES Email Sending Errors (p. 423).
After Amazon SES Sends an Email

If the sender's request to Amazon SES succeeds, then Amazon SES sends the email and one of the following outcomes occurs:

- **Successful delivery and the recipient does not object to the email**—The email is accepted by the ISP, and the ISP delivers the email to the recipient. A successful delivery is shown in the following figure.

- **Hard bounce**—The email is rejected by the ISP because of a persistent condition or rejected by Amazon SES because the email address is on the Amazon SES suppression list. An email address is on the Amazon SES suppression list if it has recently caused a hard bounce for any Amazon SES customer. A hard bounce with an ISP can occur because the recipient's address is invalid. A hard bounce notification is sent from the ISP back to Amazon SES, which notifies the sender through email or through Amazon Simple Notification Service (Amazon SNS), depending on the sender's setup. Amazon SES notifies the sender of suppression list bounces by the same means. The path of a hard bounce from an ISP is shown in the following figure.

- **Soft bounce**—The ISP cannot deliver the email to the recipient because of a temporary condition, such as the ISP is too busy to handle the request or the recipient's mailbox is full. A soft bounce can also occur if the domain does not exist. The ISP sends a soft bounce notification back to Amazon SES, or, in the case of a nonexistent domain, Amazon SES cannot find an email server for the domain. In either case, Amazon SES retries the email for an extended period of time. If Amazon SES cannot deliver the email in that time period, it sends you a bounce notification through email or through Amazon SNS. If Amazon SES can deliver the email to the recipient during a retry, the delivery is successful. A soft bounce is shown in the following figure. In this case, Amazon SES retries sending the email, and the ISP is eventually able to deliver it to the recipient.

- **Complaint**—The email is accepted by the ISP and delivered to the recipient, but the recipient considers the email to be spam and clicks a button such as "Mark as spam" in his or her email client. If Amazon SES has a feedback loop set up with the ISP, then a complaint notification is sent to Amazon SES, which forwards the complaint notification to the sender. Most ISPs do not provide the email address of the recipient who submitted the complaint, so the complaint notification from Amazon SES provides the sender a list of recipients who might have sent the complaint, based on the recipients of the original message and the ISP from which Amazon SES received the complaint. The path of a complaint is shown in the following figure.

- **Auto response**—The email is accepted by the ISP, and the ISP delivers it to the recipient. The ISP then sends an automatic response such as an out-of-the-office (OOTO) message to Amazon SES. Amazon SES forwards the auto response notification to the sender. An auto response is shown in the following figure.
Make sure that your Amazon SES-enabled program does not retry sending messages that generate an auto response.

**Tip**
You can use the Amazon SES mailbox simulator to test a successful delivery, bounce, complaint, OOTO, or what happens when an address is on the suppression list. For more information, see Testing Amazon SES Email Sending (p. 162).

### Email Format and Amazon SES

When a client makes a request to Amazon SES, Amazon SES constructs an email message compliant with the Internet Message Format specification ([RFC 5322](https://tools.ietf.org/html/rfc5322)). An email consists of a header, a body, and an envelope, as described below.

- **Header**—Contains routing instructions and information about the message. Examples are the sender’s address, the recipient’s address, the subject, and the date. The header is analogous to the information at the top of a postal letter, though it can contain many other types of information, such as the format of the message.

- **Body**—Contains the text of the message itself.

- **Envelope**—Contains the actual routing information that is communicated between the email client and the mail server during the SMTP session. This email envelope information is analogous to the information on a postal envelope. The routing information of the email envelope is usually the same as the routing information in the email header, but not always. For example, when you send a blind carbon copy (BCC), the actual recipient address (derived from the envelope) is not the same as the "To" address that is displayed in the recipient's email client, which is derived from the header.

The following is a simple example of an email. The header is followed by a blank line and then the body of the email. The envelope isn’t shown because it is communicated between the client and the mail server during the SMTP session, rather than a part of the email itself.

```
Received: from abc.smtp-out.amazonses.com (123.45.67.89) by in.example.com (87.65.43.210); Fri, 17 Dec 2010 14:26:22
From: "Andrew" <andrew@example.com>
To: "Bob" <bob@example.com>
Date: Fri, 17 Dec 2010 14:26:21 -0800
Subject: Hello
Message-ID: <61967230-7A45-4A9D-BEC9-87CBCF2211C9@example.com>
Accept-Language: en-US
Content-Language: en-US
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0

Hello, I hope you are having a good day.

-Andrew
```

The following sections review email headers and bodies and identify the information that you need to provide when you use Amazon SES.
Email Header

There is one header per email message. Each line of the header contains a field followed by a colon followed by a field body. When you read an email in an email client, the email client typically displays the values of the following header fields:

- **To**—The email addresses of the message's recipients.
- **CC**—The email addresses of the message's carbon copy recipients.
- **From**—The email address from which the email is sent.
- **Subject**—A summary of the message topic.
- **Date**—The time and date the email is sent.

There are many additional header fields that provide routing information and describe the content of the message. Email clients typically do not display these fields to the user. For a full list of the header fields that Amazon SES accepts, see Appendix: Header Fields (p. 455). When you use Amazon SES, you particularly need to understand the difference between "From," "Reply-To," and "Return-Path" header fields. As noted previously, the "From" address is the email address of the message sender, whereas "Reply-To" and "Return-Path" are as follows:

- **Reply-To**—The email address to which replies will be sent. By default, replies are sent to the original sender's email address.
- **Return-Path**—The email address to which message bounces and complaints should be sent. "Return-Path" is sometimes called "envelope from," "envelope sender," or "MAIL FROM."

**Note**
When you use Amazon SES, we recommend that you always set the "Return-Path" parameter so that you can be aware of bounces and take corrective action if they occur.

To easily match a bounced message with its intended recipient, you can use Variable Envelope Return Path (VERP). With VERP, you set a different "Return-Path" for each recipient, so that if the message bounces back, you automatically know which recipient it bounced from, rather than having to open the bounce message and parse it.

Email Body

The email body contains the text of the message. The body can be sent in the following formats:

- **HTML**—If the recipient's email client can interpret HTML, the body can include formatted text and hyperlinks
- **Plain text**—If the recipient's email client is text-based, the body must not contain any nonprintable characters.
- **Both HTML and plain text**—When you use both formats to send the same content in a single message, the recipient's email client decides which to display, based upon its capabilities.

If you are sending an email message to a large number of recipients, then it makes sense to send it in both HTML and text. Some recipients will have HTML-enabled email clients, so that they can click embedded hyperlinks in the message. Recipients using text-based email clients will need you to include URLs that they can copy and open using a web browser.

Email Information You Need to Provide to Amazon SES

When you send an email with Amazon SES, the email information you need to provide depends on how you call Amazon SES. You can provide a minimal amount of information and have Amazon SES take care of all of the formatting for you. Or, if you want to do something more advanced like send an attachment,
you can provide the raw message yourself. The following sections review what you need to provide when you send an email by using the Amazon SES API, the Amazon SES SMTP interface, or the Amazon SES console.

**Amazon SES API**

If you call the Amazon SES API directly, you call either the SendEmail or the SendRawEmail API. The amount of information you need to provide depends on which API you call.

- The SendEmail API requires you to provide only a source address, destination address, message subject, and a message body. You can optionally provide "Reply-To" addresses. When you call this API, Amazon SES automatically assembles a properly formatted multi-part Multipurpose Internet Mail Extensions (MIME) email message optimized for display by email client software. For more information, see Sending Formatted Email Using the Amazon SES API (p. 100).

- The SendRawEmail API provides you the flexibility to format and send your own raw email message by specifying headers, MIME parts, and content types. SendRawEmail is typically used by advanced users. You need to provide the body of the message and all header fields that are specified as required in the Internet Message Format specification (RFC 5322). For more information, see Sending Raw Email Using the Amazon SES API (p. 100).

If you use an AWS SDK to call the Amazon SES API, you provide the information listed above to the corresponding functions (for example, SendEmail and SendRawEmail for Java).

For more information about sending email using the Amazon SES API, see Using the Amazon SES API to Send Email (p. 99).

**Amazon SES SMTP Interface**

When you access Amazon SES through the SMTP interface, your SMTP client application assembles the message, so the information you need to provide depends on the application you are using. At a minimum, the SMTP exchange between a client and a server requires a source address, a destination address, and message data. If you are using the SMTP interface and have feedback forwarding enabled, then your bounces, complaints, and delivery notifications are sent to the "MAIL FROM" address. Any "Reply-To" address that you specify is not used.

For more information about sending email using the Amazon SES SMTP interface, see Using the Amazon SES SMTP Interface to Send Email (p. 66).

**Amazon SES Console**

When you send an email by using the Amazon SES console, the amount of information you need to provide depends on whether you choose to send a formatted or raw email.

- To send a formatted email, you need to provide a source address, a destination address, a message subject, and a message body. Amazon SES automatically assembles a properly formatted multi-part MIME email message optimized for display by email client software. You can also specify a reply-to and a return path field.

- To send a raw email, you provide the source address, a destination address, and the message content, which must contain the body of the message and all header fields that are specified as required in the Internet Message Format specification (RFC 5322).

**Getting Started Sending Email with Amazon SES**

This getting started tutorial provides step-by-step instructions for you to set up Amazon Simple Email Service (Amazon SES) and send an email. First, review the information in Before You Begin with Amazon SES.
Then, send an email in one of the following ways. You can also watch our Getting Started with Amazon SES video.

For information about Amazon SES email pricing, see the Amazon SES Pricing page.

Using the Amazon SES Console

Use this method if you want to get started sending test emails through Amazon SES with minimal setup. When you are ready to start your production email sending campaign, use one of the other sending methods and use the Amazon SES console primarily to monitor your sending activity.

To start this tutorial, go to Send an Email Using the Amazon SES Console (p. 18).

Using Simple Mail Transfer Protocol (SMTP)

Use this method if you want to send email through the Amazon SES SMTP interface with or without programming as follows:

- Enable an application to send email through Amazon SES by using a programming language that supports SMTP. Examples are provided in C#, Java, and PHP. To start this tutorial, go to Send an Email by Accessing the Amazon SES SMTP Interface Programmatically (p. 19).
- Set up your mail server to forward mail to Amazon SES, or configure your email client or SMTP-enabled software package to send email through Amazon SES. Examples are provided for Postfix, Sendmail, and Exim mail servers, as well as email client Microsoft Outlook and issue-tracking software Jira. To start this tutorial, go to Configuring Your Existing Email Server or SMTP-Enabled Application to Send Email Through Amazon SES (p. 28).

For introductory information on both SMTP sending methods, see Send an Email Through Amazon SES Using SMTP (p. 19).

Using an AWS SDK

Use this method to call the Amazon SES API using libraries that handle the details of the underlying Amazon SES Query interface. Examples are provided in C#, Java, PHP, Ruby, and Python. To start this tutorial, go to Send an Email Through Amazon SES Using an AWS SDK (p. 28).

Before You Begin with Amazon SES

Before you start, you need to set up Amazon SES. Whether you send an email by using the Amazon SES console, the SMTP interface, or the Amazon SES API, you need to:

- **Sign up for AWS**—Before you can use Amazon SES or other AWS services, you need to create an AWS account. For information, see Signing up for AWS (p. 42).
- **Verify your email address or domain**—To send emails using Amazon SES, you always need to verify your "From" address to show that you own it. If you are in the sandbox, you also need to verify your "To" addresses. You can verify email addresses or entire domains. For information, see Verifying Identities in Amazon SES (p. 43).

This list contains the setup tasks that are mandatory for all email sending methods. Additional setup tasks that are specific to the email sending method are provided in the corresponding getting started section. To see a complete list of all setup tasks, see Setting up Email with Amazon SES (p. 42).
Send an Email Using the Amazon SES Console

The easiest way to send an email with Amazon SES is to use the Amazon SES console. Because the console requires you to manually enter information, you typically only use it to send test emails. After you get started with Amazon SES, you will most likely send your emails using either the Amazon SES SMTP interface or API, but the console is useful for monitoring your sending activity.

Important
In this getting started tutorial, you send an email to yourself so that you can check to see if you received it. For further experimentation or load testing, use the Amazon SES mailbox simulator. Emails that you send to the mailbox simulator do not count toward your sending quota or your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 162).

Before you follow these steps, make sure you review the setup instructions in Before You Begin with Amazon SES (p. 17).

To send an email message from the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.

   Note
   If you are not currently signed in to your AWS account, this link takes you to a sign-in page. After you sign in, you are directed to the Amazon SES console.

2. In the navigation pane on the left side of the Amazon SES console, under Identity Management, choose Email Addresses to view the email address that you verified in Verifying Email Addresses in Amazon SES (p. 43).

3. In the list of identities, check the box next to email address that you have verified.

4. Choose Send a Test Email.

5. For Send Test Email, choose the Email Format. The two choices are as follows:

   - **Formatted**—This is the simplest option. Choose this option if you simply want to type the text of your message into the Body text box. When you send the email, Amazon SES puts the text into email format for you.

   - **Raw**—Choose this option if you want to send a more complex message, such as a message that includes HTML or an attachment. Because of this flexibility, you need to format the message, as described in Sending Raw Email Using the Amazon SES API (p. 100), yourself, and then paste the entire formatted message, including the headers, into the Body text box. You can use the following example, which contains HTML, to send a test email using the Raw email format. Copy and paste this message in its entirety into the Body text box. Ensure that there is not a blank line between the MIME-Version header and the Content-Type header; a blank line between these two lines causes the email to be formatted as plain text instead of HTML.

```plaintext
Subject: Amazon SES Raw Email Test
MIME-Version: 1.0
Content-Type: text/html

<!DOCTYPE html>
<html>
<body>
<h1>This text should be large, because it is formatted as a header in HTML.</h1>
</body>
</html>
```
6. For **Send Test Email**, fill out the rest of the fields. If you are still in the Amazon SES sandbox, make sure that the address in the **To** field is a verified email address. For more information, see [Verifying Email Addresses in Amazon SES](#) (p. 43).

7. Choose **Send Test Email**.

8. Sign in to the email client of the address you sent the email to. You will find the message that you sent.

## Send an Email Through Amazon SES Using SMTP

To send an email using the Amazon SES SMTP interface, you can use an SMTP-enabled programming language, email server, or application. Before you start, review the instructions in [Before You Begin with Amazon SES](#) (p. 17). You also need to get the following additional information:

- Your Amazon SES SMTP username and password, which enable you to connect to the Amazon SES SMTP endpoint. To get your Amazon SES SMTP username and password, see [Obtaining Your Amazon SES SMTP Credentials](#) (p. 67).

**Important**

Your SMTP credentials are different from your AWS credentials. For more information about credentials, see [Using Credentials With Amazon SES](#) (p. 369).

- The Amazon SES SMTP hostname, which is `email-smtp.us-east-1.amazonaws.com` (for Region `us-east-1`), `email-smtp.us-west-2.amazonaws.com` (for Region `us-west-2`), or `email-smtp.eu-west-1.amazonaws.com` (for Region `eu-west-1`).

- The Amazon SES SMTP interface port number, which depends on the connection method. For more information, see [Connecting to the Amazon SES SMTP Endpoint](#) (p. 70).

After you get your SMTP credentials, you can connect to the Amazon SES SMTP endpoint and send email. This getting started tutorial shows you how to send email through the Amazon SES SMTP interface by using the following methods:

- **Send an Email by Accessing the Amazon SES SMTP Interface Programmatically** (p. 19)
- **Configuring Your Existing Email Server or SMTP-Enabled Application to Send Email Through Amazon SES** (p. 28)

For more information about the Amazon SES SMTP interface, see [Using the Amazon SES SMTP Interface to Send Email](#) (p. 66).

## Send an Email by Accessing the Amazon SES SMTP Interface Programmatically

You can access the Amazon SES SMTP interface by using an SMTP-enabled programming language. You provide the Amazon SES SMTP hostname and port number along with your SMTP credentials and then use the programming language's generic SMTP functions to send the email.

**Topics in this section:**

- **Send an Email Using SMTP with C#** (p. 20)
- **Send an Email Using SMTP with Java** (p. 22)
- **Send an Email Using SMTP with PHP** (p. 25)
Send an Email Using SMTP with C#

The following procedure shows how to use Microsoft Visual Studio to create a C# console application that sends an email through Amazon SES. The procedures in this section apply to Visual Studio 2017, but the process of creating C# console applications is similar across Microsoft Visual Studio editions.

Before you perform the following procedure, complete the setup tasks described in Before You Begin with Amazon SES (p. 17) and Send an Email Through Amazon SES Using SMTP (p. 19).

Important
In this getting started tutorial, you send an email to yourself so that you can check to see if you received it. For further experimentation or load testing, use the Amazon SES mailbox simulator. Emails that you send to the mailbox simulator do not count toward your sending quota or your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 162).

To send an email using the Amazon SES SMTP interface with C#

1. Create a console project in Visual Studio by performing the following steps:
   a. Open Microsoft Visual Studio.
   c. On the New Project window, in the left pane, expand Installed, expand Templates, and then expand Visual C#.
   e. On the menu at the top of the window, choose .NET Framework 4.5, as shown in the following image.
   g. In the Name field, type AmazonSESSample.
   h. Choose OK.

2. In your Visual Studio project, replace the entire contents of Program.cs with the following code:

```csharp
using System;
using System.Net;
using System.Net.Mail;

namespace AmazonSESSample
{
    class Program
    {
        static void Main(string[] args)
        {
            // Replace sender@example.com with your "From" address.
            // This address must be verified with Amazon SES.
            const string FROM = "sender@example.com";
            
```
const String FROMNAME = "Sender Name";

// Replace recipient@example.com with a "To" address. If your account // is still in the sandbox, this address must be verified.
const String TO = "recipient@example.com";

// Replace smtp_username with your Amazon SES SMTP user name.
const String SMTP_USERNAME = "smtp_username";

// Replace smtp_password with your Amazon SES SMTP user name.
const String SMTP_PASSWORD = "smtp_password";

// (Optional) the name of a configuration set to use for this message. // If you comment out this line, you also need to remove or comment out // the "X-SES-CONFIGURATION-SET" header below.
const String CONFIGSET = "ConfigSet";

// If you're using Amazon SES in a region other than US West (Oregon), // replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP // endpoint in the appropriate Region.
const String HOST = "email-smtp.us-west-2.amazonaws.com";

// The port you will connect to on the Amazon SES SMTP endpoint. We // are choosing port 587 because we will use STARTTLS to encrypt // the connection.
const int PORT = 587;

// The subject line of the email
const String SUBJECT = "Amazon SES test (SMTP interface accessed using C#)";

// The body of the email
const String BODY = "<h1>Amazon SES Test</h1>" + 
  "<p>This email was sent through the " + 
  "<a href='https://aws.amazon.com/ses'>Amazon SES</a> SMTP interface " + 
  "using the .NET System.Net.Mail library.";</n>

// Create and build a new MailMessage object
MailMessage message = new MailMessage();
message.IsBodyHtml = true;
message.From = new MailAddress(FROM, FROMNAME);
message.To.Add(new MailAddress(TO));
message.Subject = SUBJECT;
message.Body = BODY;
// Comment or delete the next line if you are not using a configuration set
message.Headers.Add("X-SES-CONFIGURATION-SET", CONFIGSET);

// Create and configure a new SmtpClient
SmtpClient client = 
  new SmtpClient(HOST, PORT);
// Pass SMTP credentials
client.Credentials = 
  new NetworkCredential(SMTP_USERNAME, SMTP_PASSWORD);
// Enable SSL encryption
client.EnableSsl = true;

// Send the email.
try
{
  Console.WriteLine("Attempting to send email...");
  client.Send(message);
  Console.WriteLine("Email sent!");
}
catch (Exception ex)
{
}
3. In Program.cs, replace the following email addresses with your own values:

   **Important**
   The email addresses are case-sensitive. Make sure that the addresses are exactly the same as the ones you verified.

   - SENDER@EXAMPLE.COM—Replace with your "From" email address. You must verify this address before you run this program. For more information, see Verifying Identities in Amazon SES (p. 43).
   - RECIPIENT@EXAMPLE.COM—Replace with your "To" email address. If your account is still in the sandbox, you must verify this address before you use it. For more information, see Moving Out of the Amazon SES Sandbox (p. 64).

4. In Program.cs, replace the following SMTP credentials with the values that you obtained in Obtaining Your Amazon SES SMTP Credentials (p. 67):

   **Important**
   Your SMTP credentials are different from your AWS credentials. For more information about credentials, see Using Credentials With Amazon SES (p. 369).

   - YOUR_SMTP_USERNAME—Replace with your SMTP username. Note that your SMTP username credential is a 20-character string of letters and numbers, not an intelligible name.
   - YOUR_SMTP_PASSWORD—Replace with your SMTP password.

5. (Optional) If you want to use an Amazon SES SMTP endpoint in a Region other than US West (Oregon), change the value of the variable HOST to the endpoint you want to use. For a list of Amazon SES endpoints, see Regions and Amazon SES (p. 409).


7. To build the project, choose Build and then choose Build Solution.

8. To run the program, choose Debug and then choose Start Debugging.

9. Review the output. If the email was successfully sent, the console displays "Email sent!" Otherwise, it displays an error message.

10. Sign in to the email client of the recipient address. You will find the message that you sent.

### Send an Email Using SMTP with Java

This example uses the Eclipse IDE and the JavaMail API to send email through Amazon SES using the SMTP interface.

Before you perform the following procedure, complete the setup tasks described in Before You Begin with Amazon SES (p. 17) and Send an Email Through Amazon SES Using SMTP (p. 19).

**Important**
In this getting started tutorial, you send an email to yourself so that you can check to see if you received it. For further experimentation or load testing, use the Amazon SES mailbox simulator. Emails that you send to the mailbox simulator do not count toward your sending quota or your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 162).
To send an email using the Amazon SES SMTP interface with Java

1. In a web browser, go to the JavaMail Github page. Under Downloads, choose javax.mail.jar to download the latest version of JavaMail.

   **Important**
   This tutorial requires JavaMail version 1.5 or later.

2. Create a project in Eclipse by performing the following steps:
   a. Start Eclipse.
   b. In Eclipse, choose File, choose New, and then choose Java Project.
   c. In the Create a Java Project dialog box, type a project name and then choose Next.
   d. In the Java Settings dialog box, choose the Libraries tab.
   e. Choose Add External JARs.
   f. Browse to the folder in which you downloaded JavaMail. Choose the file javax.mail.jar, and then choose Open.
   g. In the Java Settings dialog box, choose Finish.

3. In Eclipse, in the Package Explorer window, expand your project.

4. Under your project, right-click the src directory, choose New, and then choose Class.

5. In the New Java Class dialog box, in the Name field, type AmazonSESSample and then choose Finish.

6. Replace the entire contents of AmazonSESSample.java with the following code:

   ```java
   package AmazonSESSample;
   import java.util.*;
   import javax.mail.*;
   import javax.mail.internet.*;
   
   public class AmazonSESSample {
     // Replace sender@example.com with your "From" address.
     // This address must be verified.
     static final String FROM = "sender@example.com";
     static final String FROMNAME = "Sender Name";
     // Replace recipient@example.com with a "To" address. If your account
     // is still in the sandbox, this address must be verified.
     static final String TO = "recipient@example.com";
     // Replace smtp_username with your Amazon SES SMTP user name.
     static final String SMTP_USERNAME = "smtp_username";
     // Replace smtp_password with your Amazon SES SMTP password.
     static final String SMTP_PASSWORD = "smtp_password";
     // The name of the Configuration Set to use for this message.
     // If you comment out or remove this variable, you will also need to
     // comment out or remove the header below.
     static final String CONFIGSET = "ConfigSet";
     // Amazon SES SMTP host name. This example uses the US West (Oregon) Region.
     static final String HOST = "email-smtp.us-west-2.amazonaws.com";
     // The port you will connect to on the Amazon SES SMTP endpoint.
     static final int PORT = 587;
     static final String SUBJECT = "Amazon SES test (SMTP interface accessed using Java)";
   }
   ```
static final String BODY = String.join(
  System.getProperty("line.separator"),
  "<h1>Amazon SES SMTP Email Test</h1>",
  "<p>This email was sent with Amazon SES using the ",
  "<a href='https://github.com/javaee/javamail'>Javamail Package</a> ",
  ");

public static void main(String[] args) throws Exception {

    // Create a Properties object to contain connection configuration information.
    Properties props = System.getProperties();
    props.put("mail.transport.protocol", "smtp");
    props.put("mail.smtp.port", PORT);
    props.put("mail.smtp.starttls.enable", "true");
    props.put("mail.smtp.auth", "true");

    // Create a Session object to represent a mail session with the specified
    // properties.
    Session session = Session.getDefaultInstance(props);

    // Create a message with the specified information.
    MimeMessage msg = new MimeMessage(session);
    msg.setFrom(new InternetAddress(FROM, FROMNAME));
    msg.setRecipient(Message.RecipientType.TO, new InternetAddress(TO));
    msg.setSubject(SUBJECT);
    msg.setContent(BODY, "text/html");

    // Add a configuration set header. Comment or delete the
    // next line if you are not using a configuration set
    msg.setHeader("X-SES-CONFIGURATION-SET", CONFIGSET);

    // Create a transport.
    Transport transport = session.getTransport();

    // Send the message.
    try {
        System.out.println("Sending... ");

        // Connect to Amazon SES using the SMTP username and password you specified
        above.
        transport.connect(HOST, SMTP_USERNAME, SMTP_PASSWORD);

        // Send the email.
        transport.sendMessage(msg, msg.getAllRecipients());
        System.out.println("Email sent!");
    }
    catch (Exception ex) {
        System.out.println("The email was not sent.");
        System.out.println("Error message: " + ex.getMessage());
    }
    finally {
        // Close and terminate the connection.
        transport.close();
    }
}

7. In AmazonSESSample.java, replace the following email addresses with your own values:
Important
The email addresses are case-sensitive. Make sure that the addresses are exactly the same as the ones you verified.

- SENDER@EXAMPLE.COM—Replace with your "From" email address. You must verify this address before you run this program. For more information, see Verifying Identities in Amazon SES (p. 43).
- RECIPIENT@EXAMPLE.COM—Replace with your "To" email address. If your account is still in the sandbox, you must verify this address before you use it. For more information, see Moving Out of the Amazon SES Sandbox (p. 64).

8. In AmazonSESSample.java, replace the following SMTP credentials with the values that you obtained in Obtaining Your Amazon SES SMTP Credentials (p. 67):

Important
Your SMTP credentials are different from your AWS credentials. For more information about credentials, see Using Credentials With Amazon SES (p. 369).

- YOUR_SMTP_USERNAME—Replace with your SMTP username credential. Note that your SMTP username credential is a 20-character string of letters and numbers, not an intelligible name.
- YOUR_SMTP_PASSWORD—Replace with your SMTP password.

9. (Optional) If you want to use an Amazon SES SMTP endpoint in a Region other than US West (Oregon), change the value of the variable HOST to the endpoint you want to use. For a list of Amazon SES endpoints, see Regions and Amazon SES (p. 409).

10. Save AmazonSESSample.java.

11. To build the project, choose Project and then choose Build Project. (If this option is disabled, then you may have automatic building enabled.)

12. To start the program and send the email, choose Run and then choose Run again.

13. Review the output. If the email was successfully sent, the console displays "Email sent!" Otherwise, it displays an error message.

14. Sign into the email client of the recipient address. You will find the message that you sent.

Send an Email Using SMTP with PHP

This example uses the PHPMailer package to send email through Amazon SES using the SMTP interface.

Important
In this tutorial, you send an email to yourself so that you can check to see if you received it. For further experimentation or load testing, use the Amazon SES mailbox simulator. Emails that you send to the mailbox simulator do not count toward your sending quota or your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 162).

Prerequisites

Before you begin, perform the following tasks:

- Verify your email address with Amazon SES—Before you can send an email with Amazon SES, you must verify that you own the sender's email address. If your account is still in the Amazon SES sandbox, you must also verify the recipient email address. The easiest way to verify email addresses is by using the Amazon SES console. For more information, see Verifying Email Addresses in Amazon SES (p. 43).
- Get your SMTP credentials—You need an Amazon SES SMTP user name and password to access the Amazon SES SMTP interface. Your SMTP credentials are not the same as your AWS credentials. You can find your SMTP credentials by going to the SMTP Settings page of the Amazon SES console. For more information about SMTP credentials, see Obtaining Your Amazon SES SMTP Credentials (p. 67).
- **Install PHP**—PHP is available at [http://php.net/downloads.php](http://php.net/downloads.php). After you install PHP, add the path to PHP in your environment variables so that you can run PHP from any command prompt.

- **Install the Composer dependency manager**—The Composer dependency manager will enable you to download and install the PHPMailer class and its dependencies. To install Composer, follow the installation instructions at [https://getcomposer.org/download](https://getcomposer.org/download).

- **Install the PHPMailer package**—Once you have installed Composer, open the file `composer.json` in a text editor. In the `require` section, add the following line: "phpmailer/phpmailer": "~5.2", and then save the file. At the command line, change to the directory that contains the `composer.json` file, and then type `php composer.phar update` to download and install PHPMailer.

### Procedure

The following procedure shows how to send an email through Amazon SES with PHP.

### To send an email using the Amazon SES SMTP interface with PHP

1. Create a file named `amazon-ses-smtp-sample.php`. Open the file with a text editor and paste in the following code:

```php
<?php

// Modify the path in the require statement below to refer to the
// location of your Composer autoload.php file.
require 'path_to_sdk_inclusion';

// Instantiate a new PHPMailer
$mail = new PHPMailer;

// Tell PHPMailer to use SMTP
$mail->isSMTP();

// Replace sender@example.com with your "From" address.
// This address must be verified with Amazon SES.
$mail->setFrom('sender@example.com', 'Sender Name');

// Replace recipient@example.com with a "To" address. If your account
// is still in the sandbox, this address must be verified.
// Also note that you can include several addAddress() lines to send
// email to multiple recipients.
$mail->addAddress('recipient@example.com', 'Recipient Name');

// Replace smtp_username with your Amazon SES SMTP user name.
$mail->Username = 'smtp_username';

// Replace smtp_password with your Amazon SES SMTP password.
$mail->Password = 'smtp_password';

// Specify a configuration set. If you do not want to use a configuration
// set, comment or remove the next line.
$mail->addCustomHeader('X-SES-CONFIGURATION-SET', 'ConfigSet');

// If you're using Amazon SES in a region other than US West (Oregon),
// replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP
// endpoint in the appropriate region.
$mail->Host = 'email-smtp.us-west-2.amazonaws.com';

// The subject line of the email
$mail->Subject = 'Amazon SES test (SMTP interface accessed using PHP)';

// The HTML-formatted body of the email
$mail->Body = '<h1>Email Test</h1>
...'

?>
```
This email was sent through the Amazon SES SMTP interface using the PHPMailer class.

```php
// Tells PHPMailer to use SMTP authentication
$mail->SMTPAuth = true;

// Enable TLS encryption over port 587
$mail->SMTPSecure = 'tls';
$mail->Port = 587;

// Tells PHPMailer to send HTML-formatted email
$mail->isHTML(true);

// The alternative email body; this is only displayed when a recipient opens the email in a non-HTML email client. The \r\n represents a line break.
$mail->AltBody = "Email Test\r\nThis email was sent through the Amazon SES SMTP interface using the PHPMailer class.";

if(!$mail->send()) {
    echo "Email not sent. " , $mail->ErrorInfo , PHP_EOL;
} else {
    echo "Email sent!" , PHP_EOL;
}
```

2. In `amazon-ses-smtp-sample.php`, replace the following with your own values:

- **sender@example.com**—Replace with an email address that you have verified with Amazon SES. For more information, see Verifying Identities (p. 43). Email addresses in Amazon SES are case-sensitive. Make sure that the address you enter is exactly the same as the one you verified.

- **recipient@example.com**—Replace with the address of the recipient. If your account is still in the sandbox, you must verify this address before you use it. For more information, see Moving Out of the Amazon SES Sandbox (p. 64). Make sure that the address you enter is exactly the same as the one you verified.

- **smtp_username**—Replace with your SMTP user name credential, which you obtained from the SMTP Settings page of the Amazon SES console. This is not the same as your AWS access key ID. Note that your SMTP user name credential is a 20-character string of letters and numbers, not an intelligible name.

- **smtp_password**—Replace with your SMTP password, which you obtained from the SMTP Settings page of the Amazon SES console. This is not the same as your AWS secret access key.

- **(Optional) email-smtp.us-west-2.amazonaws.com**—If you want to use an Amazon SES SMTP endpoint in a Region other than US West (Oregon), replace this with the Amazon SES SMTP endpoint in the Region you want to use. For a list of Amazon SES SMTP endpoints, see Regions and Amazon SES (p. 409).


4. To run the program, open a command prompt in the same directory as `amazon-ses-smtp-sample.php`, and then type `php amazon-ses-smtp-sample.php`.

5. Review the output. If the email was successfully sent, the console displays "Email sent!" Otherwise, it displays an error message.

6. Sign in to the email client of the recipient address. You will find the message that you sent.
Configuring Your Existing Email Server or SMTP-Enabled Application to Send Email Through Amazon SES

You can configure your mail server, email client, or email sending software package to send messages through Amazon SES without any programming.

First, read Send an Email Through Amazon SES Using SMTP (p. 19). Then review one of the following topics, which show you how to configure a mail server to forward mail to Amazon SES:

- Integrating Amazon SES with Postfix (p. 82)
- Integrating Amazon SES with Sendmail (p. 85)
- Integrating Amazon SES with Exim (p. 94)

For information about how to configure Microsoft Outlook, an email client, to send email through Amazon SES, see Configuring Email Clients to Send Through Amazon SES (p. 71).

For information about how to configure Jira, an issue-tracking software package, to send email through Amazon SES, see Sending Email Through Amazon SES From Software Packages (p. 79).

Send an Email Through Amazon SES Using an AWS SDK

To send an email using the Amazon SES API, you can use the Query interface directly, or you can use an AWS SDK to handle low-level details such as assembling and parsing HTTP requests and responses.

Before you send email using an AWS SDK, review the instructions in Before You Begin with Amazon SES (p. 17). In order to complete the tutorials in this section, you also need to:

- Download an AWS SDK—Download and install an AWS SDK. For more information, see Downloading an AWS SDK (p. 57).
- Get your AWS credentials—To access Amazon SES programmatically, you need your AWS access keys. For more information, see Getting Your AWS Access Keys (p. 57).
- Create a shared credentials file—Follow the procedures in Create a Shared Credentials File (p. 28) to create the shared credentials file.

When you have completed the prerequisites listed above, see Send an Email through Amazon SES Programmatically using an AWS SDK (p. 29).

Create a Shared Credentials File

The following procedure shows how to create a shared credentials file in your home directory. For the SDK sample code to function properly, you must create this file.

1. In a text editor, create a new file. In the file, paste the following code:

   ```
   [default]
   aws_access_key_id = YOUR_AWS_ACCESS_KEY_ID
   aws_secret_access_key = YOUR_AWS_SECRET_ACCESS_KEY
   ```

2. In the text file you just created, replace YOUR_AWS_ACCESS_KEY_ID with your unique AWS access key ID, and replace YOUR_AWS_SECRET_ACCESS_KEY with your unique AWS secret access key.

3. Save the file. The following table shows the correct location and file name for your operating system.

---

28
If you're using...  |  Save the file as...
---|---
Windows  |  C:\Users\<yourUserName>\.aws\credentials
Linux, macOS or Unix  |  ~/.aws/credentials

**Note**
Do not include a file extension when saving the credentials file.

---

**Send an Email through Amazon SES Programmatically using an AWS SDK**

You can use an AWS SDK to send email through Amazon SES. AWS SDKs are available for several programming languages; for more information, see [Tools for Amazon Web Services](#).

**Note**
If you have not already done so, complete the prerequisites listed in [Send an Email Through Amazon SES Using an AWS SDK (p. 28)](#) before you attempt to complete the tutorials in this section.

**Topics in this section:**
- [Send an Email Using the AWS SDK for .NET (p. 29)](#)
- [Send an Email Using the AWS SDK for Java (p. 32)](#)
- [Send an Email Using the AWS SDK for PHP (p. 34)](#)
- [Send an Email Using the AWS SDK for Ruby (p. 37)](#)
- [Send an Email Using the AWS SDK for Python (Boto) (p. 39)](#)

---

**Send an Email Using the AWS SDK for .NET**

The following procedure shows you how to send an email through Amazon SES using Visual Studio and the AWS SDK for .NET.

This solution was tested using the following components:
- Microsoft Visual Studio Community 2017, version 15.4.0.
- The AWSSDK.Core package (version 3.3.18.1), installed using NuGet.
- The AWSSDK.SimpleEmail package (version 3.3.4), installed using NuGet.

**Note**
In this getting started tutorial, you send an email to yourself so that you can check to see if you received it. For further experimentation or load testing, use the Amazon SES mailbox simulator. Emails that you send to the mailbox simulator do not count toward your sending quota or your bounce and complaint rates. For more information, see [Testing Amazon SES Email Sending (p. 162)](#).

**Prerequisites**

Before you begin, perform the following tasks:
• **Verify your email address with Amazon SES**—Before you can send an email with Amazon SES, you must verify that you own the sender's email address. If your account is still in the Amazon SES sandbox, you must also verify the recipient email address. The easiest way to verify email addresses is by using the Amazon SES console. For more information, see Verifying Email Addresses in Amazon SES (p. 43).

• **Get your AWS credentials**—You need an AWS access key ID and AWS secret access key to access Amazon SES using an SDK. You can find your credentials by using the Security Credentials page of the AWS Management Console. For more information about credentials, see Using Credentials With Amazon SES (p. 369).

• **Install Visual Studio**—Visual Studio is available at https://www.visualstudio.com/.

• **Create a shared credentials file**—For the sample code in this section to function properly, you must create a shared credentials file. For more information, see Create a Shared Credentials File (p. 28).

**Procedure**

The following procedure shows how to send an email through Amazon SES using the AWS SDK for .NET.

**To send an email using the AWS SDK for .NET**

1. Create a new project by performing the following steps:
   b. On the **File** menu, choose **New, Project**.
   c. On the **New Project** window, in the panel on the left, expand **Installed**, and then expand **Visual C#**.
   d. In the panel on the right, choose **Console App (.NET Framework)**.
   e. In the **Name** field, type **AmazonSESSample**, and then choose **OK**.

2. Use NuGet to include the Amazon SES packages in your solution by completing the following steps:
   a. In Solution Explorer, right-click your project, and then choose Manage NuGet Packages for Solution.
   b. On the **NuGet - Solution** tab, choose **Browse**.
   c. In the search box, type **AWSSDK.SimpleEmail**. Choose the **AWSSDK.SimpleEmail** package.
   d. Under **Versions**, check the box next to your project, and then choose **Install**.
   e. On the **Preview Changes** window, choose **OK**.

3. On the **Program.cs** tab, paste the following code:

```csharp
using Amazon;
using System;
using System.Collections.Generic;
using Amazon.SimpleEmail;
using Amazon.SimpleEmail.Model;

namespace AmazonSESSample
{
    class Program
    {
        // Replace sender@example.com with your "From" address.
        // This address must be verified with Amazon SES.
        static readonly string senderAddress = "sender@example.com";

        // Replace recipient@example.com with a "To" address. If your account
        // is still in the sandbox, this address must be verified.
        static readonly string receiverAddress = "recipient@example.com";

        // The configuration set to use for this email. If you do not want to use a
```
// configuration set, comment out the following property and the
// ConfigurationSetName = configSet argument below.
static readonly string configSet = "ConfigSet";

// The subject line for the email.
static readonly string subject = "Amazon SES test (AWS SDK for .NET)";

// The email body for recipients with non-HTML email clients.
static readonly string textBody = "Amazon SES Test (.NET)
+ This email was sent through Amazon SES "+ using the AWS SDK for .NET.";

// The HTML body of the email.
static readonly string htmlBody = @"<html>
<head></head>
<body>
<h1>Amazon SES Test (AWS SDK for .NET)</h1>
<p>This email was sent with<br>
<a href='https://aws.amazon.com/ses/'>Amazon SES</a> using the <br>
<a href='https://aws.amazon.com/sdk-for-net/'>AWS SDK for .NET</a></p>
</body>
</html>";

static void Main(string[] args)
{
    // Replace USWest2 with the AWS Region you're using for Amazon SES.
    // Acceptable values are EUWest1, USEast1, and USWest2.
    using (var client = new AmazonSimpleEmailServiceClient(RegionEndpoint.USWest2))
    {
        var sendRequest = new SendEmailRequest
        {
            Source = senderAddress,
            Destination = new Destination
            {
                ToAddresses =
                new List<string> { receiverAddress }
            },
            Message = new Message
            {
                Subject = new Content(subject),
                Body = new Body
                {
                    Html = new Content
                    {
                        Charset = "UTF-8",
                        Data = htmlBody
                    },
                    Text = new Content
                    {
                        Charset = "UTF-8",
                        Data = textBody
                    }
                }
            },
            // If you are not using a configuration set, comment
            // or remove the following line
            ConfigurationSetName = configSet
        };
        try
        {
            Console.WriteLine("Sending email using Amazon SES... ");
            var response = client.SendEmail(sendRequest);
            Console.WriteLine("The email was sent successfully.");
        }
    }
}
catch (Exception ex)
{
    Console.WriteLine("The email was not sent.");
    Console.WriteLine("Error message: " + ex.Message);
}
}
Console.WriteLine("Press any key to continue...");
Console.ReadKey();

4. Save Program.cs.
5. Build and run the application by completing the following steps:
   a. On the Build menu, choose Build Solution.
6. Review the output of the console. If the email was successfully sent, the console displays "The email was sent successfully." Otherwise, it displays an error message.
7. If the email was successfully sent, sign in to the email client of the recipient address. You will find the message that you sent.

Send an Email Using the AWS SDK for Java

The following procedure shows you how to use Eclipse IDE for Java EE Developers and AWS Toolkit for Eclipse to create an AWS SDK project and modify the Java code to send an email through Amazon SES.

Important
In this getting started tutorial, you send an email to yourself so that you can check to see if you received it. For further experimentation or load testing, use the Amazon SES mailbox simulator. Emails that you send to the mailbox simulator do not count toward your sending quota or your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 162).

Prerequisites

Before you begin, perform the following tasks:

• **Verify your email address with Amazon SES**—Before you can send an email with Amazon SES, you must verify that you own the sender's email address. If your account is still in the Amazon SES sandbox, you must also verify the recipient email address. The easiest way to verify email addresses is by using the Amazon SES console. For more information, see Verifying Email Addresses in Amazon SES (p. 43).

• **Get your AWS credentials**—You need an AWS access key ID and AWS secret access key to access Amazon SES using an SDK. You can find your credentials by using the Security Credentials page in the AWS Management Console. For more information about credentials, see Using Credentials With Amazon SES (p. 369).

• **Install Eclipse**—Eclipse is available at https://www.eclipse.org/downloads. The code in this tutorial was tested using Eclipse Neon.3 (version 4.6.3), running version 1.8 of the Java Runtime Environment.

• **Install the AWS Toolkit for Eclipse**—Instructions for adding the AWS Toolkit for Eclipse to your Eclipse installation are available at https://aws.amazon.com/eclipse. The code in this tutorial was tested using version 2.3.1 of the AWS Toolkit for Eclipse.

• **Create a shared credentials file**—For the sample code in this section to function properly, you must create a shared credentials file. For more information, see Create a Shared Credentials File (p. 28).
Send an Email Using an AWS SDK

Procedure

The following procedure shows how to send an email through Amazon SES using the AWS SDK for Java.

To send an email using the AWS SDK for Java

1. Create an AWS Java Project in Eclipse by performing the following steps:
   a. Start Eclipse.
   b. On the File menu, choose New, and then choose Other. On the New window, expand the AWS folder, and then choose AWS Java Project.
   c. In the New AWS Java Project dialog box, do the following:
      i. For Project name, type a project name.
      ii. Under AWS SDK for Java Samples, select Amazon Simple Email Service JavaMail Sample.
      iii. Choose Finish.
2. In Eclipse, in the Package Explorer pane, expand your project.
3. Under your project, expand the src/main/java folder, expand the com.amazon.aws.samples folder, and then double-click AmazonSESSample.java.
4. Replace the entire contents of AmazonSESSample.java with the following code:

```java
package com.amazonaws.samples;
import java.io.IOException;
import com.amazonaws.services.simpleemail.*;
import com.amazonaws.services.simpleemail.model.*;
import com.amazonaws.regions.*;
public class AmazonSESSample {
    // Replace sender@example.com with your "From" address.
    // This address must be verified with Amazon SES.
    static final String FROM = "sender@example.com";
    // Replace recipient@example.com with a "To" address. If your account
    // is still in the sandbox, this address must be verified.
    static final String TO = "recipient@example.com";
    // The configuration set to use for this email. If you do not want to use a
    // configuration set, comment the following variable and the
    // .withConfigurationSetName(CONFIGSET); argument below.
    static final String CONFIGSET = "ConfigSet";
    // The subject line for the email.
    static final String SUBJECT = "Amazon SES test (AWS SDK for Java)";
    // The HTML body for the email.
    static final String HTMLBODY = "<h1>Amazon SES test (AWS SDK for Java)</h1>" + "<p>This email was sent with <a href='https://aws.amazon.com/ses/'>Amazon SES</a> using the <a href='https://aws.amazon.com/sdk-for-java/'>AWS SDK for Java</a>;"
    // The email body for recipients with non-HTML email clients.
    static final String TEXTBODY = "This email was sent through Amazon SES " + "using the AWS SDK for Java.";

    public static void main(String[] args) throws IOException {
        try {
            AmazonSimpleEmailService client =
                AmazonSimpleEmailServiceClientBuilder.standard();
```
// Replace US_WEST_2 with the AWS Region you're using for
// Amazon SES.
.withRegion(Regions.US_WEST_2).build();
SendEmailRequest request = new SendEmailRequest()
  .withDestination(new Destination().withToAddresses(TO))
  .withMessage(new Message()
    .withBody(new Body()
      .withHtml(new Content()
        .withCharset("UTF-8").withData(HTMLBODY))
      .withText(new Content()
        .withCharset("UTF-8").withData(TEXTBODY))))
  .withSource(FROM)
  // Comment or remove the next line if you are not using a
  // configuration set
  .withConfigurationSetName(CONFIGSET);
client.sendEmail(request);
System.out.println("Email sent!");
} catch (Exception ex) {
  System.out.println("The email was not sent. Error message:");
  + ex.getMessage());
}

5. In AmazonSESSample.java, replace the following with your own values:

**Important**
The email addresses are case-sensitive. Make sure that the addresses are exactly the same as the ones you verified.

- **SENDER@EXAMPLE.COM**—Replace with your "From" email address. You must verify this address before you run this program. For more information, see Verifying Identities in Amazon SES (p. 43).
- **RECIPIENT@EXAMPLE.COM**—Replace with your "To" email address. If your account is still in the sandbox, you must verify this address before you use it. For more information, see Moving Out of the Amazon SES Sandbox (p. 64).
- **US_WEST_2**—Set this to the AWS Region of the Amazon SES endpoint you want to connect to. Note that your sandbox status, sending limits, and Amazon SES identity-related settings are specific to a given AWS Region, so be sure to select an AWS Region in which you set up Amazon SES. In this example, we are using the US West (Oregon) Region. Examples of other Regions that Amazon SES supports are US_EAST_1 and EU_WEST_1. For a complete list of AWS Regions that Amazon SES supports, see Regions and Amazon SES (p. 409).

6. Save AmazonSESSample.java.
7. To build the project, choose Project and then choose Build Project.

**Note**
If this option is disabled, automatic building may be enabled; if so, skip this step.

8. To start the program and send the email, choose Run and then choose Run again.
9. Review the output of the console pane in Eclipse. If the email was successfully sent, the console displays "Email sent!" Otherwise, it displays an error message.
10. If the email was successfully sent, sign in to the email client of the recipient address. You will find the message that you sent.

**Send an Email Using the AWS SDK for PHP**

This topic shows how to use the AWS SDK for PHP to send an email through Amazon SES.
Important
In this tutorial, you send an email to yourself so that you can check to see if you received it. For further experimentation or load testing, use the Amazon SES mailbox simulator. Emails that you send to the mailbox simulator do not count toward your sending quota or your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 162).

Prerequisites
Before you begin, perform the following tasks:

- **Verify your email address with Amazon SES**—Before you can send an email with Amazon SES, you must verify that you own the sender’s email address. If your account is still in the Amazon SES sandbox, you must also verify the recipient email address. The easiest way to verify email addresses is by using the Amazon SES console. For more information, see Verifying Email Addresses in Amazon SES (p. 43).
- **Get your AWS credentials**—You need an AWS access key ID and AWS secret access key to access Amazon SES using an SDK. You can find your credentials by using the Security Credentials page of the AWS Management Console. For more information about credentials, see Using Credentials With Amazon SES (p. 369).
- **Install PHP**—PHP is available at http://php.net/downloads.php. This tutorial requires PHP version 5.5 or higher. After you install PHP, add the path to PHP in your environment variables so that you can run PHP from any command prompt. The code in this tutorial was tested using PHP 7.0.20.
- **Install the AWS SDK for PHP version 3**—For download and installation instructions, see the AWS SDK for PHP documentation. The code in this tutorial was tested using version 3.31.0 of the SDK.
- **Create a shared credentials file**—For the sample code in this section to function properly, you must create a shared credentials file. For more information, see Create a Shared Credentials File (p. 28).

Procedure
The following procedure shows how to send an email through Amazon SES using the AWS SDK for PHP.

To send an email through Amazon SES using the AWS SDK for PHP

1. In a text editor, create a file named amazon-ses-sample.php. Paste the following code:

```php
<?php
// Replace path_to_sdk_inclusion with the path to the SDK as described in
define('REQUIRED_FILE', 'path_to_sdk_inclusion');

// Replace sender@example.com with your "From" address.
// This address must be verified with Amazon SES.
define('SENDER', 'sender@example.com');

// Replace recipient@example.com with a "To" address. If your account
// is still in the sandbox, this address must be verified.
define('RECIPIENT', 'recipient@example.com');

// Specify a configuration set. If you do not want to use a configuration
// set, comment the following variable, and the
// 'ConfigurationSetName' => CONFIGSET argument below.
define('CONFIGSET', 'ConfigSet');

// Replace us-west-2 with the AWS Region you're using for Amazon SES.
define('REGION', 'us-west-2');

define('SUBJECT', 'Amazon SES test (AWS SDK for PHP)');
```
define('HTMLBODY','<h1>AWS Amazon Simple Email Service Test Email</h1>
  This email was sent with <a href="https://aws.amazon.com/ses/">Amazon SES</a> using the <a href="https://aws.amazon.com/sdk-for-php/">AWS SDK for PHP</a>.
'.';
define('TEXTBODY','This email was send with Amazon SES using the AWS SDK for PHP.');

define('CHARSET','UTF-8');

require REQUIRED_FILE;

use Aws\Ses\SesClient;
use Aws\Ses\Exception\SesException;

$client = SesClient::factory(array(
  'version' => 'latest',
  'region' => REGION
));

try {
  $result = $client->sendEmail(
    'Destination' => [
      'ToAddresses' => [
        RECIPIENT,
      ],
    ],
    'Message' => [
      'Body' => [
        'Html' => [
          'Charset' => CHARSET,
          'Data' => HTMLBODY,
        ],
        'Text' => [
          'Charset' => CHARSET,
          'Data' => TEXTBODY,
        ],
      ],
      'Subject' => [
        'Charset' => CHARSET,
        'Data' => SUBJECT,
      ],
    ],
    'Source' => SENDER,
    // If you are not using a configuration set, comment or delete the following line
    // 'ConfigurationSetName' => CONFIGSET,
  ));
  $messageId = $result->get('MessageId');
  echo("Email sent! Message ID: $messageId\n");
} catch (SesException $error) {
  echo("The email was not sent. Error message: ".$error->getAwsErrorMessage()."\n");
}

2. In amazon-ses-sample.php, replace the following with your own values:

- **path_to_sdk_inclusion**—Replace with the path required to include the AWS SDK for PHP in the program. For more information, see the AWS SDK for PHP documentation.

- **sender@example.com**—Replace with an email address that you have verified with Amazon SES. For more information, see Verifying Identities (p. 43). Email addresses in Amazon SES are case-sensitive. Make sure that the address you enter is exactly the same as the one you verified.
• recipient@example.com—Replace with the address of the recipient. If your account is still in the sandbox, you must verify this address before you use it. For more information, see Moving Out of the Amazon SES Sandbox (p. 64). Make sure that the address you enter is exactly the same as the one you verified.

• (Optional) us-west-2—If you want to use Amazon SES in a Region other than US West (Oregon), replace this with the Region you want to use. For a list of Regions in which Amazon SES is available, see Regions and Amazon SES (p. 409).


4. To run the program, open a command prompt in the same directory as amazon-ses-sample.php, and then type `php amazon-ses-sample.php`

5. Review the output. If the email was successfully sent, the console displays "Email sent!" Otherwise, it displays an error message.

   **Note**
   If you encounter a "cURL error 60: SSL certificate problem" error when you run the program, download the latest CA bundle as described in the AWS SDK for PHP documentation. Then, in amazon-ses-sample.php, add the following lines to the SesClient::factory array, replace path_of_certs with the path to the CA bundle you downloaded, and re-run the program.

   ```
   'http' => [
     'verify' => 'path_of_certs\ca-bundle.crt'
   ]
   ```

6. Sign in to the email client of the recipient address. You will find the message that you sent.

Send an Email Using the AWS SDK for Ruby

This topic shows how to use the AWS SDK for Ruby to send an email through Amazon SES.

**Important**
In this tutorial, you send an email to yourself so that you can check to see if you received it. For further experimentation or load testing, use the Amazon SES mailbox simulator. Emails that you send to the mailbox simulator do not count toward your sending quota or your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 162).

**Prerequisites**
Before you begin, perform the following tasks:

• **Verify your email address with Amazon SES**—Before you can send an email with Amazon SES, you must verify that you own the sender's email address. If your account is still in the Amazon SES sandbox, you must also verify the recipient email address. The easiest way to verify email addresses is by using the Amazon SES console. For more information, see Verifying Email Addresses in Amazon SES (p. 43).

• **Get your AWS credentials**—You need an AWS access key ID and AWS secret access key to access Amazon SES using an SDK. You can find your credentials by using the Security Credentials page of the AWS Management Console. For more information about credentials, see Using Credentials With Amazon SES (p. 369).

• **Install Ruby**—Ruby is available at https://www.ruby-lang.org/en/downloads/. The code in this tutorial was tested using Ruby 1.9.3. After you install Ruby, add the path to Ruby in your environment variables so that you can run Ruby from any command prompt.

• **Install the AWS SDK for Ruby**—For download and installation instructions, see Installing the AWS SDK for Ruby in the AWS SDK for Ruby Developer Guide. The sample code in this tutorial was tested using version 2.9.36 of the AWS SDK for Ruby.
• **Create a shared credentials file**—For the sample code in this section to function properly, you must create a shared credentials file. For more information, see Create a Shared Credentials File (p. 28).

**Procedure**

The following procedure shows how to send an email through Amazon SES using the AWS SDK for Ruby.

**To send an email through Amazon SES using the AWS SDK for Ruby**

1. In a text editor, create a file named `amazon-ses-sample.rb`. Paste the following code into the file:

```ruby
require 'aws-sdk'

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
sender = "sender@example.com"

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
recipient = "recipient@example.com"

# Specify a configuration set. If you do not want to use a configuration
# set, comment the following variable and the
# configuration_set_name: configsetname argument below.
configsetname = "ConfigSet"

# Replace us-west-2 with the AWS Region you're using for Amazon SES.
awsregion = "us-west-2"

# The subject line for the email.
subject = "Amazon SES test (AWS SDK for Ruby)"

# The HTML body of the email.
htmlbody = "<h1>Amazon SES test (AWS SDK for Ruby)</h1>
'Amazon SES</a> using the <a href="https://aws.amazon.com/sdk-for-ruby/">AWS SDK for Ruby</a>.'

# The text body for recipients with non-HTML email clients.
textbody = "This email was sent with Amazon SES using the AWS SDK for Ruby."

# Specify the text encoding scheme.
encoding = "UTF-8"

# Create a new SES resource and specify a region
ses = Aws::SES::Client.new(region: awsregion)

# Try to send the email.
begin
  # Provide the contents of the email.
  resp = ses.send_email({
    destination: {
      to_addresses: [
        recipient,
      ],
    },
    message: {
      body: {
        html: {
          charset: encoding,
          data: htmlbody,
        }
      }
    }
  })
end
```
Send an Email Using the AWS SDK for Python (Boto)

This topic shows how to use the AWS SDK for Python (Boto) to send an email through Amazon SES.

Important
In this tutorial, you send an email to yourself so that you can check to see if you received it. For further experimentation or load testing, use the Amazon SES mailbox simulator. Emails that you send to the mailbox simulator do not count toward your sending quota or your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 162).

Prerequisites
Before you begin, perform the following tasks:

- **Verify your email address with Amazon SES**—Before you can send an email with Amazon SES, you must verify that you own the sender's email address. If your account is still in the Amazon SES sandbox, you must verify this address before you use it. For more information, see Moving Out of the Amazon SES Sandbox (p. 64). Make sure that the address you enter is exactly the same as the one you verified.

2. In `amazon-ses-sample.rb`, replace the following with your own values:

- `sender@example.com`—Replace with an email address that you have verified with Amazon SES. For more information, see Verifying Identities (p. 43). Email addresses in Amazon SES are case-sensitive. Make sure that the address you enter is exactly the same as the one you verified.

- `recipient@example.com`—Replace with the address of the recipient. If your account is still in the sandbox, you must verify this address before you use it. For more information, see Moving Out of the Amazon SES Sandbox (p. 64). Make sure that the address you enter is exactly the same as the one you verified.

- **(Optional) us-west-2**—If you want to use Amazon SES in a region other than US West (Oregon), replace this with the region you want to use. For a list of regions in which Amazon SES is available, see Regions and Amazon SES (p. 409).


4. To run the program, open a command prompt in the same directory as `amazon-ses-sample.rb`, and type `ruby amazon-ses-sample.rb`

5. Review the output. If the email was successfully sent, the console displays "Email sent!" Otherwise, it displays an error message.

6. Sign in to the email client of the recipient address. You will find the message that you sent.
sandbox, you must also verify the recipient email address. The easiest way to verify email addresses is by using the Amazon SES console. For more information, see Verifying Email Addresses in Amazon SES (p. 43).

- **Get your AWS credentials**—You need an AWS access key ID and AWS secret access key to access Amazon SES using an SDK. You can find your credentials by using the Security Credentials page of the AWS Management Console. For more information about credentials, see Using Credentials With Amazon SES (p. 369).

- **Install Python**—Python is available at https://www.python.org/downloads/. The code in this tutorial was tested using Python 2.7.6 and Python 3.6.1. After you install Python, add the path to Python in your environment variables so that you can run Python from any command prompt.

- **Install the AWS SDK for Python (Boto)**—For download and installation instructions, see the AWS SDK for Python (Boto) documentation. The sample code in this tutorial was tested using version 1.4.4 of the SDK for Python.

- **Create a shared credentials file**—For the sample code in this section to function properly, you must create a shared credentials file. For more information, see Create a Shared Credentials File (p. 28).

**Procedure**

The following procedure shows how to send an email through Amazon SES using the SDK for Python.

**To send an email through Amazon SES using the SDK for Python**

1. In a text editor, create a file named `amazon-ses-sample.py`. Paste the following code into the file:

```python
import boto3
from botocore.exceptions import ClientError

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
# SENDER = "Sender Name <sender@example.com>"

# Replace recipient@example.com with a "To" address. If your account is still in the sandbox, this address must be verified.
# RECIPIENT = "recipient@example.com"

# Specify a configuration set. If you do not want to use a configuration set, comment the following variable, and the ConfigurationSetName=CONFIGURATION_SET argument below.
# CONFIGURATION_SET = "ConfigSet"

# If necessary, replace us-west-2 with the AWS Region you're using for Amazon SES.
# AWS_REGION = "us-west-2"

# The subject line for the email.
SUBJECT = "Amazon SES Test (SDK for Python)"

# The email body for recipients with non-HTML email clients.
BODY_TEXT = "This email was sent with Amazon SES using the "
           "AWS SDK for Python (Boto)."

# The HTML body of the email.
BODY_HTML = """"<html>
<head></head>
<body>
<h1>Amazon SES Test (SDK for Python)</h1>
<p>This email was sent with
   <a href='https://aws.amazon.com/ses/'>Amazon SES</a> using the
   <a href='https://aws.amazon.com/sdk-for-python/'>AWS SDK for Python (Boto)</a>
</body></html>"
```

40
AWS SDK for Python (Boto)<a>.</a></p>  
</body>  
</html>

```python
# The character encoding for the email.
CHARSET = "UTF-8"

# Create a new SES resource and specify a region.
client = boto3.client('ses',region_name=AWS_REGION)

# Try to send the email.
try:
    # Provide the contents of the email.
    response = client.send_email(
        Destination={
            'ToAddresses': [RECIPIENT],
        },
        Message={
            'Body': {
                'Html': {
                    'Charset': CHARSET,
                    'Data': BODY_HTML,
                },
                'Text': {
                    'Charset': CHARSET,
                    'Data': BODY_TEXT,
                },
            },
            'Subject': {
                'Charset': CHARSET,
                'Data': SUBJECT,
            },
        },
        Source=SENDER,
        # If you are not using a configuration set, comment or delete the
        # following line
        ConfigurationSetName=CONFIGURATION_SET,
    )
    # Display an error if something goes wrong.
    except ClientError as e:
        print(e.response['Error']['Message'])
    else:
        print("Email sent! Message ID:")
        print(response['ResponseMetadata']['RequestId'])
```

2. In `amazon-ses-sample.py`, replace the following with your own values:

- **sender@example.com**—Replace with an email address that you have verified with Amazon SES. For more information, see Verifying Identities (p. 43). Email addresses in Amazon SES are case-sensitive. Make sure that the address you enter is exactly the same as the one you verified.

- **recipient@example.com**—Replace with the address of the recipient. If your account is still in the sandbox, you must verify this address before you use it. For more information, see Moving Out of the Amazon SES Sandbox (p. 64). Make sure that the address you enter is exactly the same as the one you verified.

- **(Optional) us-west-2**—If you want to use Amazon SES in a region other than US West (Oregon), replace this with the region you want to use. For a list of regions in which Amazon SES is available, see Regions and Amazon SES (p. 409).


41
4. To run the program, open a command prompt in the same directory as `amazon-ses-sample.py`, and then type `python amazon-ses-sample.py`
5. Review the output. If the email was successfully sent, the console displays "Email sent!" Otherwise, it displays an error message.
6. Sign in to the email client of the recipient address. You will find the message that you sent.

Setting up Email with Amazon SES

To set up email with Amazon Simple Email Service (Amazon SES), you need to perform the following tasks:

- Before you can access Amazon SES or other AWS services, you need to set up an AWS account. For more information, see Signing up for AWS (p. 42).
- Before you send email through Amazon SES, you need to verify that you own the "From" address. If your account is still in the Amazon SES sandbox, you also need to verify your "To" addresses. You can verify email addresses or entire domains. For more information, see Verifying Identities in Amazon SES (p. 43).

The following tasks are optional depending on what you want to do:

- If you want to access Amazon SES through the Amazon SES API, whether by the Query (HTTPS) interface or indirectly through an AWS SDK, the AWS Command Line Interface or the AWS Tools for Windows PowerShell, you need to obtain your AWS access keys. For more information, see Getting Your AWS Access Keys (p. 57).
- If you want to call the Amazon SES API without handling the low-level details of the Query interface, you can use an AWS SDK. For more information, see Downloading an AWS SDK (p. 57).
- If you want to access Amazon SES through its SMTP interface, you need to obtain your SMTP username and password. Your SMTP credentials are different from your AWS credentials. For more information, see Getting Your SMTP Credentials for Amazon SES (p. 64).
- When you first sign up for Amazon SES, your account is in the Amazon SES sandbox. In the sandbox, you can send emails using the same email-sending methods as any other Amazon SES user, except that you can only send 200 emails per 24-hour period at a maximum rate of one email per second, and you can only send emails to addresses you have verified. To increase your sending limits and to send email to unverified email addresses, see Moving Out of the Amazon SES Sandbox (p. 64).
- If you want your emails to pass Domain-based Message Authentication, Reporting and Conformance (DMARC) authentication based on Sender Policy Framework (SPF), configure your identity to send from a custom MAIL FROM domain as described in Using a Custom MAIL FROM Domain (p. 57).

Signing up for AWS

You need to create an AWS account before you can use Amazon SES or other AWS services. When you create an AWS account, AWS automatically signs up your account for all services. You are charged only for the services that you use.

**Note**

If you will be sending your emails from an Amazon EC2 instance either directly or through AWS Elastic Beanstalk, you can get started with Amazon SES for free. For more information, see Amazon SES Pricing.

When you first sign up for AWS, your Amazon SES sending is in the Amazon SES sandbox. In the sandbox, you have full access to the Amazon SES API and SMTP interface. However, the following restrictions are in effect:
• You can only send emails to the Amazon SES mailbox simulator and to email addresses or domains that you have verified. For more information, see Verifying Identities in Amazon SES (p. 43).
• You can send a maximum of 200 messages per 24-hour period.
• You can send a maximum of one message per second.

For information about moving out of the sandbox, see Moving Out of the Amazon SES Sandbox (p. 64).

To create an AWS account
1. Go to https://aws.amazon.com/ses, and choose Sign up now.
2. Follow the on-screen instructions.

Note
Even if your account is out of the Amazon SES sandbox, you still need to verify your "From" address to confirm that you own it.

Verifying Identities in Amazon SES

In Amazon SES, an identity is an email address or domain that you use to send email. Before you can send an email using Amazon SES, you must verify each identity that you’re going to use as a "From", "Source", "Sender", or "Return-Path" address to prove that you own it. If your account is still in the Amazon SES sandbox, you also need to verify any email addresses that you send emails to, except for email addresses provided by the Amazon SES mailbox simulator (p. 162).

You can verify an identity by using the Amazon SES console or the Amazon SES API.

Verifying Email Addresses in Amazon SES

Amazon SES requires that you verify your identities (the domains or email addresses that you send email from) to confirm that you own them, and to prevent unauthorized use. This section includes information about verifying email address identities. For information about verifying domain identities, see the section called “Verifying Domains” (p. 52).

Consider the following factors when you verify email addresses for use with Amazon SES:

• You must verify each identity that you use as a "From," "Source," "Sender," or "Return-Path" address. You can, however, add a label to an email address that has already been verified without performing any additional verification steps (see the information later in this list).
• Email addresses are case sensitive. If you verify sender@example.com, you can’t send email from sender@example.com unless you verify sender@example.com as well.
• If you verify both an email address and the domain that address belongs to, the settings for the email address override those of the domain. For example, if DomainKeys Identified Mail (DKIM) is enabled for the domain example.com, but not for sender@example.com, emails sent from sender@example.com aren’t DKIM signed.
• Amazon SES has endpoints in multiple AWS Regions, and the verification status of the email address is separate for each region. If you want to send email from the same identity in more than one region, you must verify that identity in each region. For information about using Amazon SES in multiple regions, see Regions (p. 409).
• You can verify up to 10,000 identities across your entire AWS account.
• You can add labels to verified email addresses without performing additional verification steps. To add a label to an email address, add a plus sign (+) between the account name and the "at" sign (@), followed by a text label. For example, if you already verified sender@example.com, you can use sender
+myLabel@example.com as the "From" or "Return-Path" address for your emails. You can use this feature to implement Variable Envelope Return Path (VERP). Then you can use VERP to detect and remove undeliverable email addresses from your mailing lists.

- You can customize the messages that are sent to the email addresses you attempt to verify. For more information, see the section called “Using Custom Verification Email Templates” (p. 47).

## Verifying an Email Address

You can verify email addresses by using the Amazon SES console or the VerifyEmailIdentity API operation.

### Verifying an Email Address Using the Amazon SES Console

Complete the procedure in this section to verify an email address using the Amazon SES console.

#### To verify an email address using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the console, use the region selector to choose the AWS Region where you want to verify the email address, as shown in the following image.

   ![Amazon SES Console](image)

   **Note**
   To verify an email address for use in more than one region, repeat the procedure in this section for each region.

3. In the navigation pane, under Identity Management, choose Email Addresses.
4. Choose Verify a New Email Address.
5. In the Verify a New Email Address dialog box, type your email address in the Email Address field, and then choose Verify This Email Address.
6. Check the inbox for the email address that you're verifying. You'll receive a message with the following subject line: "Amazon Web Services - Email Address Verification Request in region RegionName," where RegionName is the name of the AWS Region you selected in step 2.

   Click the link in the message.

   **Note**
   The link in the verification message expires 24 hours after the message was sent. If 24 hours have passed since you received the verification email, repeat steps 1–5 to receive a verification email with a valid link.

7. In the Amazon SES console, under Identity Management, choose Email Addresses. In the list of email addresses, locate the email address you're verifying. If the email address was verified, the value in the Status column is “verified”.

---

44
Verify an Email Address Using the Amazon SES API

Use the VerifyEmailIdentity API operation to create a new email identity. When you execute this operation, a verification email is sent to the specified address.

To verify an email address using the AWS CLI, type the following command at the command line: `aws ses verify-email-identity --email-address sender@example.com`

In the preceding command, replace `sender@example.com` with the email address that you want to verify.

For a script that can be used to verify several email identities in a single operation, see the section called “Verify Multiple Email Addresses” (p. 407).

Troubleshoot Email Address Verification

If you attempted to verify an email address, but didn't receive a verification email from AWS, try the following troubleshooting steps:

- Check the Junk Mail folder in your email client.
- Ensure that your email client isn't applying rules that automatically move certain messages to a folder other than your inbox.
- In your email client, add no-reply-aws@amazon.com to your address book or Safe Senders list. You can also ask your system administrator to whitelist incoming email from no-reply-aws@amazon.com.
- With an email address that uses a different email service provider (such as a personal email address), send a message to the address you want to verify. Ensure that the address you want to verify receives the message. This step is especially important if you recently set up your own domain. Occasionally, new domains might not be correctly configured to receive incoming email.

Alternatively, try to verify an email address that you know is able to receive email, such as a personal email address. If you receive the verification email at your personal address, it might indicate that there is an issue on the other domain.

If these tests show email isn't being received at the address you attempted to verify, consult your system administrator or email service provider for further assistance.

Listing Email Identities in Amazon SES

You can display a list of email identities by using the Amazon SES console or the ListIdentities API operation.

Viewing a List of Email Identities in Amazon SES

You can use the Amazon SES console and API to view a list of email addresses that are verified or are pending verification, as well as those that failed the verification process.

To view a list of verified email addresses

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the console, use the region selector to choose the AWS Region where you want to list email identities, as shown in the following image.
Verifying Identities

3. In the navigation pane, under Identity Management, choose Email Addresses.

The Email Addresses page displays a list of email addresses that are verified, that are pending verification, and that failed the verification process. Click an email address to view additional information about it.

Viewing a List of Email Identities Using the Amazon SES API

Use the ListIdentities API operation to view a list of all email identities, regardless of their statuses. You can also use the GetIdentityVerificationAttributes operation to find the verification status of a given identity.

To view a list of identities by using the AWS CLI, type the following command at the command line: `aws ses list-identities`

When you execute the ListIdentities operation, it returns a list of all of the identities in your Amazon SES account, regardless of their verification statuses. To see the verification status for one or more identities, use the GetIdentityVerificationAttributes operation. To find the verification status of an identity using the AWS CLI, type the following command at the command line: `aws ses get-identity-verification-attributes --identities "sender@example.com"

Replace `sender@example.com` in the preceding command with the identity that you want to find the verification status of. You can also use this command to find the verification statuses of multiple identities in a single API call. For example, to find the status of the domain `example.com` and the email address `sender@example.co.uk`, type the following command: `aws ses get-identity-verification-attributes --identities "example.com" "sender@example.co.uk"

Deleting an Email Identity in Amazon SES

If you no longer need to use a verified email address, you can delete it by using the Amazon SES console or the DeleteIdentity API operation.

Warning
This action can't be undone. However, you can repeat the verification process for an identity that was previously deleted.

Deleting an Email Identity in Amazon SES

To remove verified email addresses

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the console, use the region selector to choose the AWS Region where you want to delete an email identity, as shown in the following image.

![AWS Console](image1)

**Note**
These procedures only delete the email address in the selected AWS Region. To delete an email address that was verified in more than one region, repeat the procedure in this section for each region.

3. Select each email address that you want to remove, and then choose **Remove**.

### Deleting an Email Identity Using the Amazon SES API

Use the `DeleteIdentity` API operation to delete email address and domain identities.

To delete an identity using the AWS CLI, type the following command at the command line:

```
aws ses delete-identity --identity "sender@example.com"
```

Replace `sender@example.com` in the preceding command with the identity that you want to delete.

### Using Custom Verification Email Templates

When you attempt to verify an email address, Amazon SES sends an email to that address that resembles the example shown in the following image.

![Example Verification Email](image2)

Several Amazon SES customers build applications (such as email marketing suites or ticketing systems) that send email through Amazon SES on behalf of their own customers. For the end users of these applications, the email verification process can be confusing: the verification email uses Amazon SES branding, rather than the branding of the application, and those end users never signed up to use Amazon SES directly.
If your Amazon SES use case requires your customers to have their email addresses verified for use with Amazon SES, you can create customized verification emails. These customized emails help reduce customer confusion and increase the rates at which your customers complete the registration process.

Topics in this section:
- Creating a Custom Verification Email Template (p. 48)
- Editing a Custom Verification Email Template (p. 49)
- Sending Verification Emails Using Custom Templates (p. 49)
- Custom Verification Email Frequently Asked Questions (p. 50)

Creating a Custom Verification Email Template

To create a custom verification email, use the CreateCustomVerificationEmailTemplate API operation. This operation takes the following inputs:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TemplateName</td>
<td>The name of the template. The name you specify must be unique.</td>
</tr>
<tr>
<td>FromEmailAddress</td>
<td>The email address that the verification email is sent from. The address or domain you specify must be verified for use with your Amazon SES account.</td>
</tr>
<tr>
<td>TemplateSubject</td>
<td>The subject line of the verification email.</td>
</tr>
<tr>
<td>TemplateContent</td>
<td>The body of the email. The email body can contain HTML, with certain limitations. For more information, see Custom Verification Email Frequently Asked Questions (p. 50).</td>
</tr>
<tr>
<td>SuccessRedirectionURL</td>
<td>The URL that users are sent to if their email addresses are successfully verified.</td>
</tr>
<tr>
<td>FailureRedirectionURL</td>
<td>The URL that users are sent to if their email addresses are not successfully verified.</td>
</tr>
</tbody>
</table>

You can use the AWS SDKs or the AWS CLI to create a custom verification email template with the CreateCustomVerificationEmailTemplate operation. To learn more about the AWS SDKs, see Tools for Amazon Web Services. For more information about the AWS CLI, see AWS Command Line Interface.

The following section includes procedures for creating a custom verification email using the AWS CLI. These procedures assume that you have installed and configured the AWS CLI. For more information about installing and configuring the AWS CLI, see the AWS Command Line Interface User Guide.

1. In a text editor, create a new file. Paste the following content into the editor:

```json
{
    "TemplateName": "SampleTemplate",
    "FromEmailAddress": "sender@example.com",
    "TemplateSubject": "Please confirm your email address",
    "TemplateContent": "<html>
    <head></head>
    <body style="font-family:sans-serif;">
    <h1 style="text-align:center">Ready to start sending email with ProductName?</h1>
    <p>We here at Example Corp are happy to have you on board! There’s just one last step to complete before
```
you can start sending email. Just click the following link to verify your email address. Once we confirm that you’re really you, we’ll give you some additional information to help you get started with ProductName.</p>
</body>
</html>

"SuccessRedirectionURL": "https://www.example.com/verifysuccess",
"FailureRedirectionURL": "https://www.example.com/verifyfailure"
}

**Important**
To make the preceding example easier to read, the `TemplateContent` attribute contains line breaks. If you paste the preceding example into your text file, remove the line breaks before proceeding.

Replace the values of `TemplateName`, `FromEmailAddress`, `TemplateSubject`, `TemplateContent`, `SuccessRedirectionURL`, and `FailureRedirectionURL` with your own values. Save the file as `customverificationemail.json`.

2. At the command line, type the following command to create the custom verification email template: `aws ses create-custom-verification-email-template --cli-input-json file://customverificationemail.json`

3. Optionally, you can confirm that the template was created by typing the following command: `aws ses list-custom-verification-email-templates`

**Editing a Custom Verification Email Template**

You can edit a custom verification email template using the `UpdateCustomVerificationEmailTemplate` operation. This operation accepts the same inputs as the `CreateCustomVerificationEmailTemplate` operation (that is, the `TemplateName`, `FromEmailAddress`, `TemplateSubject`, `TemplateContent`, `SuccessRedirectionURL`, and `FailureRedirectionURL` attributes). However, with the `UpdateCustomVerificationEmailTemplate` operation, none of these attributes are required. When you pass a value for `TemplateName` that is the same as the name of an existing custom verification email template, the attributes you specify overwrite the attributes that were originally in the template.

**Sending Verification Emails Using Custom Templates**

After you create at least one custom verification email template, you can send it to your customers by calling the `SendCustomVerificationEmail` API operation. You can call the `SendCustomVerificationEmail` operation by using any of the AWS SDKs or the AWS CLI. The `SendCustomVerificationEmail` operation takes the following inputs:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EmailAddress</td>
<td>The email address that is being verified.</td>
</tr>
<tr>
<td>TemplateName</td>
<td>The name of the custom verification email template that is sent to email address that is being verified.</td>
</tr>
<tr>
<td>ConfigurationSetName</td>
<td>(Optional) The name of a configuration set to use when sending the verification email.</td>
</tr>
</tbody>
</table>

For example, assume your customers register for your service using a form in your application. When the customer completes the form and submits it, your application calls the `SendCustomVerificationEmail` operation, passing the customer’s email address and the name of the template you want to use.
Your customer receives an email that uses the customized email template you created. Amazon SES automatically adds a unique link to the recipient, as well as a brief disclaimer. The following image shows a sample verification email that uses the template created in the section called “Creating a Custom Verification Email Template” (p. 48).

Custom Verification Email Frequently Asked Questions

This section contains answers to frequently asked questions about the custom verification email template feature.

Q1. How many custom verification email templates can I create?

You can create up to 50 custom verification email templates per Amazon SES account.

Q2. How do custom verification emails appear to recipients?

Custom verification emails include the content you specified when you created the template, followed by a link that recipients must click to verify their email addresses.

Q3. Can I preview the custom verification email?

To preview a custom verification email, use the `SendCustomVerificationEmail` operation to send a verification email to an address you own. If you do not click the verification link, Amazon SES does not create a new identity. If you do click the verification link, you can optionally delete the newly created identity using the `DeleteIdentity` operation.

Q4. Can I include images in my custom verification email templates?

You can embed images in the HTML for your templates by using Base64 encoding. When you embed images in this way, Amazon SES automatically converts them into attachments. You can encode an image at the command line by issuing one of the following commands:

- **Linux, macOS, or Unix**:
  
  ```bash
  base64 -i imagefile.png | tr -d '\n' > output.txt
  ```

- **Windows**:
  
  ```cmd
  certutil -encode imagefile.png output.tmp && findstr /v /c:- output.tmp > output.txt && del output.tmp
  ```

Replace `imagefile.png` with the name of the file you want to encode. In both of the commands above, the Base64 encoded image is saved to `output.txt`.

**Note**

If you encoded the image using the Windows command line, you must open `output.txt` in a text editor and remove the line breaks from the file before proceeding.

You can embed the Base64-encoded image by including the following in the HTML for the template:

```html
<img src="data:image/png;base64,base64EncodedImage"/>
```

In the example above, replace `png` with the file type of the encoded image (such as jpg or gif), and replace `base64EncodedImage` with the Base64 encoded image (that is, the contents of `output.txt` from one of the preceding commands).
Q5. Are there any size limitations to the content that is included in custom verification email templates?

Custom verification email templates may not exceed 10 MB in size.

In order to prevent abuse, custom verification email templates that contain HTML can only use the tags listed in the following table:

<table>
<thead>
<tr>
<th>Allowed tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>abbr</td>
</tr>
<tr>
<td>acronym</td>
</tr>
<tr>
<td>address</td>
</tr>
<tr>
<td>area</td>
</tr>
<tr>
<td>b</td>
</tr>
<tr>
<td>bdo</td>
</tr>
<tr>
<td>big</td>
</tr>
<tr>
<td>blockquote</td>
</tr>
<tr>
<td>body</td>
</tr>
<tr>
<td>br</td>
</tr>
<tr>
<td>button</td>
</tr>
<tr>
<td>caption</td>
</tr>
<tr>
<td>center</td>
</tr>
<tr>
<td>cite</td>
</tr>
<tr>
<td>code</td>
</tr>
<tr>
<td>col</td>
</tr>
<tr>
<td>colgroup</td>
</tr>
<tr>
<td>dd</td>
</tr>
<tr>
<td>del</td>
</tr>
</tbody>
</table>

Additionally, the HTML tags in the preceding table may only include certain attributes. These attributes are listed in the following table:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Allowed attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(All allowed tags)</td>
<td>id, class, style, title</td>
</tr>
<tr>
<td>blockquote</td>
<td>cite</td>
</tr>
<tr>
<td>col</td>
<td>span, width</td>
</tr>
<tr>
<td>colgroup</td>
<td>span, width</td>
</tr>
</tbody>
</table>
Q6. How many verified email addresses can exist in my account?

Each Amazon SES account can have up to 10,000 verified identities. In Amazon SES, identities include both verified domains and email addresses. If you have verified domains or email addresses for your own email sending, those identities are included in your 10,000 identity limit.

Q7. Can I create custom verification email templates using the Amazon SES console?

Currently, it is only possible to create, edit, and delete custom verification emails using the Amazon SES API.

Q8. Can I track open and click events that occur when customers receive custom verification emails?

Custom verification emails cannot include open or click tracking.

Q9. Can custom verification emails include custom headers?

Custom verification emails cannot include custom headers.

Q10. Can I remove the text that appears at the bottom of custom verification emails?

The following text is automatically added to the end of every custom verification email and cannot be removed:

If you did not request to verify this email address, please disregard this message. If you have any concerns, please forward this message to the following email address along with your questions or comments.

The email address link in this text refers to ses-enforcement@amazon.com, an inbox that is actively monitored by the Amazon SES team.

Q11. Are custom verification emails DKIM-signed?

In order for verification emails to be DKIM-signed, the email address that you specify in the FromEmailAddress attribute when you create the verification email template must be configured to generate a DKIM signature. For more information about setting up DKIM for domains and email addresses, see the section called “Authenticating Email with DKIM” (p. 117).

Verifying Domains in Amazon SES

Amazon SES requires that you verify your email address or domain, to confirm that you own it and to prevent others from using it. When you verify an entire domain, you are verifying all email addresses from that domain, so you don't need to verify email addresses from that domain individually. For example, if you verify the domain example.com, you can send email from user1@example.com, user2@example.com, or any other user at example.com.
You can manage your verified domains by using the Amazon SES console or the Amazon SES API. For a complete description of API actions related to domain verification, go to the Amazon Simple Email Service API Reference. This section, which demonstrates the actions using the Amazon SES console, contains the following topics:

- Verifying a Domain With Amazon SES (p. 53)
- Listing Domain Identities in Amazon SES (p. 55)
- Deleting a Domain Identity in Amazon SES (p. 55)
- Amazon SES Domain Verification Revocation (p. 55)
- Amazon SES Domain Verification TXT Records (p. 56)

Important notes about domain verification are as follows:

- Amazon SES has endpoints in multiple AWS regions, and domain verification applies to each AWS region separately. You must perform the entire domain verification procedure for each region in which you want to send from a given domain. If you want to verify the same domain in multiple regions and your DNS provider does not allow you to have multiple TXT records with the same name, see the workarounds in Common Domain Verification Problems (p. 425).
- If you verify a domain with Amazon SES, you can send from any subdomain of that domain without specifically verifying the subdomain. For example, if you verify example.com, you do not need to verify a.example.com or a.b.example.com. As specified in RFC 1034, each DNS label can have up to 63 characters and the whole domain name must not exceed a total length of 255 characters.
- If you verify a domain, subdomain(s), and/or email address(es) that share a root domain, the verified identity settings (such as feedback notifications and Easy DKIM) apply at the most granular level you verified. That is:
  - Verified email address settings override verified domain settings.
  - Verified subdomain settings override verified domain settings, with lower-level subdomain settings overriding higher-level subdomain settings.

For example, assume you verify user@a.b.example.com, a.b.example.com, b.example.com, and example.com. These are the verified identity settings that will be used in the following scenarios:

- Emails sent from user@example.com (an address that is not specifically verified) will use the settings for example.com.
- Emails sent from user@a.b.example.com (an address that is specifically verified) will use the settings for user@a.b.example.com.
- Emails sent from user@b.example.com (an address that is not specifically verified) will use the settings for b.example.com.
- Domain names are case-insensitive. If you verify example.com, you can send from EXAMPLE.com also.
- In each AWS region, you can verify as many as 10,000 identities (domains and email addresses, in any combination) per AWS account.

Verifying a Domain With Amazon SES

The following procedure shows you how to verify a domain using the Amazon SES console. If you want to use the Amazon SES API instead, see the Amazon Simple Email Service API Reference.

To verify a domain

1. Go to your verified domain list in the Amazon SES console, or follow these instructions to navigate to it:
   a. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
b. In the navigation pane, under Identity Management, choose Domains.

2. Choose Verify a New Domain.

3. In the Verify a New Domain dialog box, enter the domain name. If you want to set up DKIM signing for this domain, select the Generate DKIM Settings option. (For information about DKIM signing, see Authenticating Email with DKIM in Amazon SES (p. 117).) Choose Verify This Domain.

4. In the Verify a New Domain dialog box, you will see a Domain Verification Record Set containing a Name, a Type, and a Value. (This information will also be available by choosing the domain name after you close the dialog box.)

To complete domain verification, add a TXT record with the displayed Name and Value to your domain's DNS server. For information about Amazon SES TXT records and general guidance about how to add a TXT record to a DNS server, see Amazon SES Domain Verification TXT Records (p. 56). In particular:

- If your DNS provider does not allow underscores in record names, you can omit _amazonses from the Name.
- To help you easily identify this record within your domain's DNS settings, you can optionally prefix the Value with amazonses:
- Some DNS providers automatically append the domain name to DNS record names. To avoid duplication of the domain name, you can add a period to the end of the domain name in the DNS record. This indicates that the record name is fully qualified and the DNS provider need not append an additional domain name.

5. If Route 53 provides the DNS service for the domain that you are verifying, and you are signed in to the AWS Management Console under the same account that you use for Route 53, then Amazon SES will give you the option of updating your DNS server immediately from within the Amazon SES console. If you are not using Route 53, Amazon SES needs to verify that a TXT record with the specified Name and Value have been added to your domain's DNS server. This may take up to 72 hours.

When verification is complete, the domain's status in the Amazon SES console will change from "pending verification" to "verified," and you will receive a confirmation success email from Amazon SES to the email address associated with your AWS account.
6. You can now use Amazon SES to send email from any address in the verified domain. To send a test email, check the box next to the verified domain, and then choose **Send a Test Email**.

If the DNS settings are not correctly updated, you will receive a domain verification failure email from Amazon SES, and the domain will display a status of "failed" in the **Domains** tab. If this happens, read our troubleshooting page at Amazon SES Email Address and Domain Verification Problems (p. 424). When you have verified that your TXT record is correctly in place, choose the "retry" link next to the "failed" status notification. This will reinitiate the domain verification process.

**Listing Domain Identities in Amazon SES**

To view your verified domains, follow the procedure below.

**To view your verified domains**

1. Go to your verified domain list in the Amazon SES console, or follow these instructions to navigate to it:
   a. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
   b. In the navigation pane, under **Identity Management**, choose **Domains**.
2. In the list of verified domains, you can expand one or more domains to view the details.

**Deleting a Domain Identity in Amazon SES**

To remove a verified domain, follow the procedure below.

**To remove a verified domain**

1. Go to your verified domain list in the Amazon SES console, or follow these instructions to navigate to it:
   a. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
   b. In the navigation pane, under **Identity Management**, choose **Domains**.
2. Check the box beside each domain that you want to remove, and then choose **Remove**.
3. You will no longer be able to send email from the removed domain.

**Amazon SES Domain Verification Revocation**

Amazon SES periodically reviews domain verification status, and revokes verification in cases where it is no longer valid. If Amazon SES is unable to detect the TXT record information required to confirm ownership of a domain, you will receive an **Amazon SES Domain Verification REVOCATION WARNING** email from Amazon SES.

If you restore the TXT record information to your domain's DNS server within 72 hours, you will receive an **Amazon SES Domain Verification REVOCATION CANCELLATION** email from Amazon SES.

**Note**

You can review the required TXT record information in the Amazon SES console by using the following instructions. In the navigation pane, under **Identity Management**, choose **Domains**. In the list of domains, choose (not just expand) the domain to display the domain verification settings, which include the TXT record name and value.

If you do not restore the TXT record information to your domain's DNS server within 72 hours, you will receive an **Amazon SES Domain Verification REVOCATION** email from Amazon SES, the domain will be...
removed from the list of **Verified Senders** on the **Domains** tab, and you will no longer be able to send from the domain.

To reverify a domain for which verification has been revoked, you must restart the verification procedure from the beginning, just as if the revoked domain were an entirely new domain.

**Amazon SES Domain Verification TXT Records**

Your domain is associated with a set of Domain Name System (DNS) records that you manage through your DNS provider. A TXT record is a type of DNS record that provides additional information about your domain. Each TXT record consists of a name and a value.

When you initiate domain verification using the Amazon SES console or API, Amazon SES gives you the name and value to use for the TXT record. For example, if your domain is `example.com`, the TXT record settings that Amazon SES generates will look similar to the following example:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>_amazonses.example.com</td>
<td>TXT</td>
<td>pmBGN/7MjnfhTKUZ06Enqq1PeGUaOkw8lGhcfwefcHU=</td>
</tr>
</tbody>
</table>

Add a TXT record to your domain's DNS server using the specified **Name** and **Value**. Amazon SES domain verification is complete when Amazon SES detects the existence of the TXT record in your domain's DNS settings.

If your DNS provider does not allow DNS record names to contain underscores, you can omit `_amazonses` from the **Name**. In that case, for the preceding example, the TXT record name would be `example.com` instead of `_amazonses.example.com`. To make the record easier to recognize and maintain, you can also optionally prefix the **Value** with `amazonses:`. In the previous example, the value of the TXT record would therefore be `amazonses:pmBGN/7MjnfhTKUZ06Enqq1PeGUaOkw8lGhcfwefcHU=`.

**Note**

Amazon SES previously allowed TXT record names to contain `amazonses` without an underscore. If you have already verified a domain and your TXT record contains `amazonses` without an underscore, your domain will continue to be verified; there is no action required on your part. However, any new domains that you verify will require that `amazonses` in the TXT record name is either preceded by an underscore, or `_amazonses` is removed from the TXT record name entirely.

You can find troubleshooting information and instructions on how to check your domain verification settings in Amazon SES Email Address and Domain Verification Problems (p. 424).

**Adding a TXT Record to Your Domain's DNS Server**

The procedure for adding TXT records to your domain's DNS server depends on who provides your DNS service. Your DNS provider might be Route 53 or another domain name registrar such as GoDaddy. Although we cannot provide specific instructions about how to add TXT records to your domain's DNS server, here is the general procedure DNS providers typically use.

**To add a TXT record to your domain's DNS server (general procedure)**

1. Go to your DNS provider's website. If you are not sure which DNS provider serves your domain, try looking it up by using a free Whois service.
2. Sign in to your domain's account.
3. Find the page for updating your domain's DNS records. This page often has a name such as DNS Records, DNS Zone File, Advanced DNS, or something similar.
4. Locate the TXT records for your domain.
5. Add a TXT record with the name and value provided by Amazon SES.

   **Important**
   Some DNS providers automatically append the domain name to the end of DNS records. Adding a record that already contains the domain name (such as `_amazonses.example.com`) might result in the duplication of the domain name (such as `_amazonses.example.com.example.com`). To avoid duplication of the domain name, add a period to the end of the domain name in the DNS record. This will indicate to your DNS provider that the record name is fully qualified (that is, no longer relative to the domain name), and prevent the DNS provider from appending an additional domain name.

6. Save your changes. DNS record updates can take up to 48 hours to take effect, but they often take effect much sooner. You can verify that the TXT record is correctly published by using the procedure in How to Check Domain Verification Settings (p. 426).

---

**Getting Your AWS Access Keys**

After you've signed up for Amazon SES, you'll need to obtain your AWS access keys if you want to access Amazon SES through the Amazon SES API, whether by the Query (HTTPS) interface directly or indirectly through an AWS SDK, the AWS Command Line Interface, or the AWS Tools for Windows PowerShell. AWS access keys consist of an access key ID and a secret access key.

For information about getting your AWS access keys, see [AWS Security Credentials](#) in the [AWS General Reference](#).

**Downloading an AWS SDK**

If you want to call the Amazon SES API without having to handle low-level details like assembling raw HTTP requests, you can use an AWS SDK. The AWS SDKs provide functions and data types that encapsulate the functionality of Amazon SES and other AWS services. To download an AWS SDK, go to [SDKs](#).

The Getting Started section of this guide provides examples of how to send an email using the AWS SDKs for various programming languages. For more information, see [Send an Email Through Amazon SES Using an AWS SDK](#) (p. 28).

**Using a Custom MAIL FROM Domain with Amazon SES**

When an email is sent, it has two addresses that indicate its source: a From address provided in the email header, and a MAIL FROM address that the sending mail server specifies to the receiving server to indicate the source of the message. The MAIL FROM address is sometimes called the envelope sender, envelope from, bounce address, or Return Path address.

When recipients view an email in their inbox, they see the email's From address. In contrast, the MAIL FROM address, which is used by mail servers to return bounce messages and other error notifications, is typically only viewable by recipients if they inspect the email's headers in the raw message source. Amazon SES sets the MAIL FROM domain to a default value unless you choose to use your own.

**Why Use a Custom MAIL FROM Domain?**

By default, messages that you send through Amazon SES use `amazonses.com` (or a subdomain of that) as the MAIL FROM domain. Sender Policy Framework (SPF) authentication successfully validates these messages because the default MAIL FROM domain matches the sending mail server, Amazon SES. Although this level of authentication is enough for many senders, you might want to set the MAIL FROM
domain to a domain that you own to enable your emails to authenticate with Domain-based Message Authentication, Reporting and Conformance (DMARC) through SPF, which requires an additional check for SPF domain alignment. DMARC enables a sender’s domain to indicate, using a DNS record, that its emails are protected by SPF, DomainKeys Identified Mail (DKIM), or both.

There are two ways to achieve DMARC validation: using SPF and using DKIM. Unless you use your own MAIL FROM domain, you cannot achieve DMARC validation using SPF because that validation requires the domain in the From address to match the MAIL FROM domain. By using your own MAIL FROM domain, you have the flexibility to use SPF, DKIM, or both to achieve DMARC validation. For more information, see Authenticating Email with SPF (p. 116).

Choosing a MAIL FROM Domain

If you choose to use your own MAIL FROM domain with Amazon SES, the domain you use must meet the following requirements:

• The MAIL FROM domain must be a subdomain of the verified identity (email address or domain) from which you will send your emails. For example, mail.example.com is a valid MAIL FROM domain for the email address user@example.com or the domain example.com.
• In most cases, the MAIL FROM domain should not be a domain that you send email from. If you must use the MAIL FROM domain in a From address, either disable email feedback forwarding (p. 228) and receive your bounces through Amazon SNS notifications, or ensure that your MAIL FROM domain is not the destination for feedback forwarding. To determine the destination of email forwarding feedback, see Email Feedback Forwarding Destination (p. 228).
• The MAIL FROM domain should not be a domain that you use to receive email.

Setup Process

To set the MAIL FROM domain for a verified identity, you configure the verified identity using the Amazon SES console or API and publish an MX record (and optionally, an SPF record) to your MAIL FROM domain's DNS server. If at any point you want to return to using the default Amazon SES MAIL FROM domain, you can remove your MAIL FROM domain from the verified identity’s settings. These procedures are described in the following sections:

• Setting a MAIL FROM Domain (p. 58)
• Removing a MAIL FROM Domain (p. 60)
• Editing a MAIL FROM Domain (p. 61)

For a description of custom MAIL FROM domain setup states, see MAIL FROM Domain Setup States (p. 62).

Setting a MAIL FROM Domain with Amazon SES

This topic contains procedures for setting up a custom MAIL FROM domain using the Amazon SES console.

Note

You can use the same MAIL FROM address in multiple AWS Regions. For more information, see Regions and Amazon SES (p. 409).

Setting the MAIL FROM Domain

The following procedures show how to use the Amazon SES console to configure a verified email address or domain to send emails using a specified MAIL FROM domain. You can also configure a MAIL FROM
domain using the SetIdentityMailFromDomain API operation. For more information, see the Amazon Simple Email Service API Reference.

To configure a verified email address to use a specified MAIL FROM domain

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation pane, under Identity Management, choose Email Addresses.
3. In the list of verified email addresses, confirm that the status of the email address for which you want to set the MAIL FROM domain is verified. If the status is failure, choose retry and then click the link within the verification email you receive in your email client. Otherwise, choose the email address and proceed to the next step.
4. In the details pane of the verified email address, expand MAIL FROM Domain.
5. Choose Set MAIL FROM Domain.
6. In the Set MAIL FROM Domain dialog box, type the name of the MAIL FROM domain that you want to use. Note that this must be a subdomain of the domain of the verified email address, such as mail.example.com.
7. For Behavior if MX record not found, choose one of the following options:
   • Use region.amazonaws.com as MAIL FROM – If the custom MAIL FROM domain's MX record is not set up correctly, Amazon SES will use a subdomain of amazonses.com. The subdomain varies based on the AWS Region in which you use Amazon SES.
   • Reject message – If the custom MAIL FROM domain's MX record is not set up correctly, Amazon SES will return a MailFromDomainNotVerified error. Emails that you attempt to send from this address will be automatically rejected.
8. Choose Set MAIL FROM Domain. A window appears that contains the MX and SPF records that you must add to your domain's DNS configuration. Note these values, and then proceed to the next step.
9. Publish the MX record to the DNS server of the custom MAIL FROM domain.

   Important
   If the DNS configuration for the MAIL FROM domain contains multiple MX records, the custom MAIL FROM setup with Amazon SES will fail.

   a. If Route 53 provides the DNS service for your MAIL FROM domain, and you are signed in to the AWS Management Console under the same account that you use for Route 53, then choose Publish Records Using Route 53. The appropriate MX record is automatically applied to your domain's DNS configuration.
   b. If your MAIL FROM domain does not use Route 53, then you must publish the displayed MX record to the MAIL FROM domain's DNS server yourself. The procedure for adding an MX record to your domain's DNS server varies based on your web hosting service or DNS provider; see your provider's documentation for more information.

   After Amazon SES detects the record, emails you send from this verified email address will use the specified MAIL FROM domain. Until then, Amazon SES will either use the default MAIL FROM domain or reject the message, depending on the preferences you specified earlier in this procedure. Amazon SES can take up to 72 hours to detect your MX record.
10. Publish an SPF record to your MAIL FROM domain's DNS server to show receiving mail servers that you have authorized Amazon SES to send email on behalf of your domain. For more information, see Authenticating Email with SPF in Amazon SES (p. 116).

To configure a verified domain to use a specified MAIL FROM domain

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation pane, under Identity Management, choose Domains.
3. In the verified domain list, confirm that the status of the domain for which you want to set the MAIL FROM domain is verified. If the status is failure, choose retry and then add the displayed TXT record to your DNS server, as described in Amazon SES Domain Verification TXT Records (p. 56). Otherwise, choose the domain and continue this procedure.

4. In the details pane of the verified domain, expand MAIL FROM Domain.

5. Choose Set MAIL FROM Domain.

6. In the Set MAIL FROM Domain dialog box, type the name of the MAIL FROM domain that you want to use. Note that this must be a subdomain of the verified domain, such as mail.example.com.

7. For Behavior if MX record not found, choose one of the following options:
   - Use region.amazonaws.com as MAIL FROM – If the custom MAIL FROM domain's MX record is not set up correctly, Amazon SES will use a subdomain of amazonses.com. The subdomain varies based on the AWS Region in which you use Amazon SES.
   - Reject message – If the custom MAIL FROM domain's MX record is not set up correctly, Amazon SES will return a MailFromDomainNotVerified error. Emails that you attempt to send from this address will be automatically rejected.

8. Choose Set MAIL FROM Domain.

9. Next, publish an MX record to the DNS server of the custom MAIL FROM domain.

   Important
   To successfully set up a custom MAIL FROM domain with Amazon SES, you must publish exactly one MX record to the DNS server of your MAIL FROM domain. If the MAIL FROM domain has multiple MX records, the custom MAIL FROM setup with Amazon SES will fail.

   a. If Route 53 provides the DNS service for your MAIL FROM domain, and you are signed in to the AWS Management Console under the same account that you use for Route 53, then choose Publish Records Using Route 53. The appropriate MX record is automatically applied to your domain's DNS configuration.

   b. If your MAIL FROM domain does not use Route 53, then you must publish the displayed MX record to the MAIL FROM domain's DNS server yourself. The procedure for adding an MX record to your domain's DNS server varies based on your web hosting service or DNS provider; see your provider's documentation for more information.

   After Amazon SES detects the record, emails you send from this verified email address will use the specified MAIL FROM domain. Until then, Amazon SES will either use the default MAIL FROM domain or reject the message, depending on the preferences you specified earlier in this procedure. Amazon SES can take up to 72 hours to detect your MX record.

10. Publish an SPF record to your MAIL FROM domain's DNS server to show receiving mail servers that you have authorized Amazon SES to send email on behalf of your domain. For more information, see Authenticating Email with SPF in Amazon SES (p. 116).

Removing a MAIL FROM Domain with Amazon SES

If you want to use the default Amazon SES MAIL FROM domain, you can remove the custom MAIL FROM domain configuration from the verified identity.

The following procedures show how to use the Amazon SES console to remove a custom MAIL FROM domain from the configuration of a verified email address or domain. If you want to use the Amazon SES API instead, see the SetIdentityMailFromDomain API in the Amazon Simple Email Service API Reference.

To remove a custom MAIL FROM domain from the configuration of a verified email address

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation pane, under **Identity Management**, choose **Email Addresses**.
3. In the verified email address list, choose the verified email address for which you want to remove the custom MAIL FROM domain.
4. In the details pane of the verified email address, expand **MAIL FROM Domain**.
5. Choose **Remove MAIL FROM Domain**.
6. Choose **Yes, Remove MAIL FROM Domain**.
7. (Optional) Log in to your DNS service and remove the MX record that you published when you set up the MAIL FROM domain with Amazon SES.
8. (Optional) Remove the SPF record that you published when you set up the custom MAIL FROM domain with Amazon SES.

To remove a custom MAIL FROM domain from the configuration of a verified domain

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation pane, under **Identity Management**, choose **Domains**.
3. In the verified domain list, choose the verified domain for which you want to remove the custom MAIL FROM domain.
4. In the details pane of the verified domain, expand **MAIL FROM Domain**.
5. Choose **Remove MAIL FROM Domain**.
6. Choose **Yes, Remove MAIL FROM Domain**.
7. (Optional) Log in to your DNS service and remove the MX record that you published when you set up the MAIL FROM domain with Amazon SES.
8. (Optional) Remove the SPF record that you published when you set up the custom MAIL FROM domain with Amazon SES.

Editing a MAIL FROM Domain with Amazon SES

The following procedures show how to use the Amazon SES console to edit the custom MAIL FROM domain configuration of a verified email address or domain. If you want to use the Amazon SES API instead, see the `SetIdentityMailFromDomain` API in the Amazon Simple Email Service API Reference.

To edit the MAIL FROM configuration of a verified email address

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation pane, under **Identity Management**, choose **Domains**.
3. In the verified email address list, choose the email address for which you want to configure the MAIL FROM domain.
4. In the details pane of the verified email address, expand **MAIL FROM Domain**.
5. Choose **Edit MAIL FROM Domain**.
6. In the **Edit MAIL FROM Domain** dialog box, edit the settings and then choose **Save MAIL FROM Domain**.
7. If you changed the MAIL FROM domain name when you edited the settings, you must publish an MX record to the DNS server of the new MAIL FROM domain.
   a. If Route 53 provides the DNS service for your MAIL FROM domain, and you are signed in to the AWS Management Console under the same account that you use for Route 53, then choose
Publish Records Using Route 53. The appropriate MX record is automatically applied to your domain's DNS configuration.

b. If your MAIL FROM domain does not use Route 53, then you must publish the displayed MX record to the MAIL FROM domain's DNS server yourself. The procedure for adding an MX record to your domain's DNS server varies based on your web hosting service or DNS provider; see your provider's documentation for more information.

After Amazon SES detects the record, emails you send from this verified email address will use the specified MAIL FROM domain. Until then, Amazon SES will either use the default MAIL FROM domain or reject the message, depending on the preferences you specified earlier in this procedure. Amazon SES can take up to 72 hours to detect your MX record.

8. (Optional) If you changed the MAIL FROM domain name and you want Sender Policy Framework (SPF) checks to succeed, you must publish an SPF record to your MAIL FROM domain's DNS server to show receiving mail servers that you have authorized Amazon SES to send email on behalf of your domain. For more information, see Authenticating Email with SPF in Amazon SES (p. 116).

To edit the MAIL FROM configuration of a verified domain

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation pane, under Identity Management, choose Domains.
3. In the verified domain list, choose the domain for which you want to configure the MAIL FROM domain.
4. In the details pane of the verified domain, expand MAIL FROM Domain.
5. Choose Edit MAIL FROM Domain.
6. In the Edit MAIL FROM Domain dialog box, edit the settings and then choose Save MAIL FROM Domain.
7. If you changed the MAIL FROM domain name when you edited the settings, you must publish an MX record to the DNS server of the new MAIL FROM domain.
   a. If Route 53 provides the DNS service for your MAIL FROM domain, and you are signed in to the AWS Management Console under the same account that you use for Route 53, then choose Publish Records Using Route 53 if you want to publish the MX record and/or SPF record from within the Amazon SES console.
   b. If your domain does not use Route 53, then you must publish the displayed MX record to the MAIL FROM domain's DNS server yourself. The procedure for adding an MX record to your domain's DNS server depends on who provides your DNS service; please see the documentation for your DNS service. After Amazon SES detects the record, emails you send from this verified domain will use the specified MAIL FROM domain. Until then, Amazon SES will either use the default MAIL FROM domain or reject the message, depending on the preferences you specified earlier in this procedure. Amazon SES can take up to 72 hours to detect your MX record.
8. (Optional) If you changed the MAIL FROM domain name and you want Sender Policy Framework (SPF) checks to succeed, you must publish an SPF record to your MAIL FROM domain's DNS server to show receiving mail servers that you have authorized Amazon SES to send email on behalf of your domain. For more information, see Authenticating Email with SPF in Amazon SES (p. 116).

MAIL FROM Domain Setup States with Amazon SES

After you configure an identity to use a custom MAIL FROM domain, the state of the setup is "pending" while Amazon SES attempts to detect the required MX record in your DNS settings. The state then changes depending on whether Amazon SES detects the MX record. The following table describes the email-sending behavior, and the Amazon SES actions associated with each state. Each time the state changes, Amazon SES sends a notification to the email address associated with your AWS account.
<table>
<thead>
<tr>
<th>State</th>
<th>Email Sending Behavior</th>
<th>Amazon SES Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pending</td>
<td>Uses custom MAIL FROM fallback setting</td>
<td>Amazon SES attempts to detect the required MX record for 72 hours. If unsuccessful, the state changes to “Failed”.</td>
</tr>
<tr>
<td>Success</td>
<td>Uses custom MAIL FROM domain</td>
<td>Amazon SES continuously checks that the required MX record is in place.</td>
</tr>
<tr>
<td>Temporary Failure</td>
<td>Uses custom MAIL FROM fallback setting</td>
<td>Amazon SES attempts to detect the required MX record for 72 hours. If unsuccessful, the state changes to “Failed”; if successful, the state changes to “Success”.</td>
</tr>
<tr>
<td>Failed</td>
<td>Uses custom MAIL FROM fallback setting</td>
<td>Amazon SES no longer attempts to detect the required MX record. To use a custom MAIL FROM domain, you must restart the setup process in Setting a MAIL FROM Domain (p. 58).</td>
</tr>
</tbody>
</table>

**Setting up SPF Records for Amazon SES**

An SPF record indicates to ISPs that you have authorized Amazon SES to send mail for your domain. When you use Amazon SES, your decision about whether to publish an SPF record depends on whether you only require your email to pass an SPF check by the receiving mail server, or if you want your email to comply with the additional requirements needed to pass Domain-based Message Authentication, Reporting and Conformance (DMARC) authentication based on SPF. For more information, see Authenticating Email with SPF in Amazon SES (p. 116).
Getting Your SMTP Credentials for Amazon SES

To use the Amazon SES SMTP interface, you must first create an SMTP user name and password. To get your SMTP Credentials, see Obtaining Your Amazon SES SMTP Credentials (p. 67).

Important
Your SMTP user name and password are not the same as your AWS access key ID and secret access key. Do not attempt to use your AWS credentials to authenticate yourself to the Amazon SES SMTP endpoint. For more information about credentials, see Using Credentials With Amazon SES (p. 369).

Moving Out of the Amazon SES Sandbox

To help protect our customers from fraud and abuse and to help you establish your trustworthiness to ISPs and email recipients, we do not immediately grant unlimited Amazon SES usage to new users. New users are initially placed in the Amazon SES sandbox. In the sandbox, you have full access to all Amazon SES email-sending methods and features so that you can test and evaluate the service; however, the following restrictions are in effect:

- You can only send mail to the Amazon SES mailbox simulator and to verified email addresses and domains.
- You can only send mail from verified email addresses and domains.
- You can send a maximum of 200 messages per 24-hour period.
- Amazon SES can accept a maximum of one message from your account per second.

To remove the restriction on recipient addresses and increase your sending limits, you need to open a case in Support Center by using the following instructions.

To move out of the Amazon SES sandbox

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. Open an SES Sending Limits Increase case in Support Center. You can also access this link using the Dedicated IPs page in the Amazon SES console.
3. In the form, provide the following information:
   - **Region** – Choose the AWS Region for which you are requesting a sending limit increase. Note that your Amazon SES sandbox status and sending limits are separate for each AWS Region. For more information, see Regions and Amazon SES (p. 409).
   - **Limit** – Choose Desired Daily Sending Quota or Desired Maximum Send Rate. Sending limits are described in Managing Your Amazon SES Sending Limits (p. 127).

Note
The rate at which Amazon SES accepts your messages might be less than the maximum send rate.

- **New limit value** – Enter the amount you are requesting. Only request the amount you think you’ll need. Remember that you are not guaranteed to receive the amount you request, and the higher the limit you request, the more justification you need to be considered for that amount.
- **Mail Type** – Choose the type of email you plan to send using your dedicated IP address. If multiple values apply, choose the type that will make up the majority of your email sending.
- **Website URL** – Type the URL of your website. Providing this information will help us better understand the type of content you plan to send.
- **My email sending complies with the AWS Service Terms and AUP** – Choose the option that applies to your use case.
- **I only send to recipients who have specifically requested my mail** – Choose the option that applies to your use case.
- **I have a process to handle bounces and complaints** – Choose the option that applies to your use case.
- **Use Case Description** – Describe the ways in which you will use Amazon SES to send email. To help us process your request more quickly, please answer the following questions:
  - How will you build or acquire your mailing list?
  - How will you handle bounces and complaints?
  - How can recipients unsubscribe from your mailing list, and how will you respond to those requests?
  - How did you choose the increase amount you specified in this request?

When you finish, choose **Submit**. We will respond to the case after reviewing your request. Please allow one business day for processing.

The following are three ways to determine whether your account has been moved out of the sandbox:

- The correspondence in your SES Sending Limits Increase case indicates that your request has been granted.
- You can successfully use Amazon SES to send an email message from your verified email address to an unverified address that you own. If you receive a `MessageRejected` error instead, stating that your email address is not verified, then you are still in the sandbox.
- The Amazon SES console shows that your sending quota is higher than 200 messages per 24-hour period. To learn more, see Monitoring Your Amazon SES Sending Limits (p. 128).

Once you are out of the sandbox, you no longer have to verify "To" addresses or domains; however, you must still verify any additional "From" or "Return-Path" addresses or domains. Over time, Amazon SES will gradually increase your sending limits, or you can open another SES Sending Limits Increase case if the gradual increase does not meet your needs. For more information, see Managing Your Amazon SES Sending Limits (p. 127).

## Sending Your Email with Amazon SES

You can send an email with Amazon Simple Email Service (Amazon SES) using the Amazon SES console, the Amazon SES Simple Mail Transfer Protocol (SMTP) interface, or the Amazon SES API. You typically use the console to send test emails and manage your sending activity. To send bulk emails, you use either the SMTP interface or the API. For information about Amazon SES email pricing, see Amazon SES Pricing.

- If you want to use an SMTP-enabled software package, application, or programming language to send email through Amazon SES, or integrate Amazon SES with your existing mail server, use the Amazon SES SMTP interface. For more information, see Using the Amazon SES SMTP Interface to Send Email (p. 66).
- If you want to call Amazon SES by using raw HTTP requests, use the Amazon SES API. For more information, see Using the Amazon SES API to Send Email (p. 99).

Before you send emails, see Setting up Email with Amazon SES (p. 42).

**Important**

When you send an email to multiple recipients (recipients are "To", "CC", and "BCC" addresses) and the call to Amazon SES fails, the entire email is rejected and none of the recipients will
receive the intended email. We therefore recommend that you send an email to one recipient at a time.

Using the Amazon SES SMTP Interface to Send Email

To send production email through Amazon SES, you can use the Simple Mail Transfer Protocol (SMTP) interface or the Amazon SES API. For more information about the Amazon SES API, see Using the Amazon SES API to Send Email (p. 99). This section describes the SMTP interface.

Amazon SES sends email using the SMTP, the most common email protocol on the Internet. You can send email through Amazon SES by using a variety of SMTP-enabled programming languages and software to connect to the Amazon SES SMTP interface. This section explains how to get your Amazon SES SMTP credentials, how to send email by using the SMTP interface, and how to configure several pieces of software and mail servers to use Amazon SES for email sending.

Note
For solutions to common problems that you might encounter when you use Amazon SES through its SMTP interface, see Amazon SES SMTP Issues (p. 430).

To send email using the Amazon SES SMTP interface, you will need the following:

- An AWS account. For more information, see Signing up for AWS (p. 42).
- The SMTP interface hostname (i.e., endpoint). For a list of Amazon SES SMTP endpoints, see Connecting to the Amazon SES SMTP Endpoint (p. 70).
- The SMTP interface port number. The port number varies with the connection method. For more information, see Connecting to the Amazon SES SMTP Endpoint (p. 70).
- An SMTP user name and password. You can use the same set of SMTP credentials in all AWS regions.

  Important
  Your SMTP user name and password are not identical to your AWS access keys or the credentials you use to log into the Amazon SES console. For information about how to generate your SMTP user name and password, see Obtaining Your Amazon SES SMTP Credentials (p. 67).

- Client software that can communicate using Transport Layer Security (TLS). For more information, see Connecting to the Amazon SES SMTP Endpoint (p. 70).
- An email address that you have verified with Amazon SES. For more information, see Verifying Identities in Amazon SES (p. 43).
- Higher sending limits, if you want to send large quantities of email. For more information, see Managing Your Amazon SES Sending Limits (p. 127).

Then, you can send email by doing the following:

- To configure an email client to send email through Amazon SES, including an example for Microsoft Outlook, see Configuring Email Clients to Send Through Amazon SES (p. 71).
- To configure SMTP-enabled software to send email through the Amazon SES SMTP interface, including an example for issue-tracking software Jira, see Sending Email Through Amazon SES From Software Packages (p. 79).
- To program an application to send email through Amazon SES, see Sending Email Through Amazon SES From Your Application (p. 81).
- To configure your existing email server to send all of your outgoing mail through Amazon SES, see Integrating Amazon SES with Your Existing Email Server (p. 81).
- To interact with the Amazon SES SMTP interface using the command line, which can be useful for testing, see Using the Command Line to Send Email Through the Amazon SES SMTP Interface (p. 95).
Email Information to Provide

When you access Amazon SES through the SMTP interface, your SMTP client application assembles the message, so the information you need to provide depends on the application you are using. At a minimum, the SMTP exchange between a client and a server requires a source address, a destination address, and message data.

If you are using the SMTP interface and have feedback forwarding enabled, then your bounces, complaints, and delivery notifications are sent to the "MAIL FROM" address. Any "Reply-To" address that you specify is not used.

Obtaining Your Amazon SES SMTP Credentials

You need an Amazon SES SMTP user name and password to access the Amazon SES SMTP interface. You can use the same set of SMTP credentials in all AWS regions.

Important

Your SMTP user name and password are not the same as your AWS access key ID and secret access key. Do not attempt to use your AWS credentials to authenticate yourself against the SMTP endpoint. For more information about credentials, see Using Credentials With Amazon SES (p. 369).

There are two ways to generate your SMTP credentials. You can either use the Amazon SES console or you can generate your SMTP credentials from your AWS credentials.

Use the Amazon SES console to generate your SMTP credentials if:

- You want to get your SMTP credentials using the simplest method.
- You do not need to automate SMTP credential generation using code or a script.

Generate your SMTP credentials from your AWS credentials if:

- You have an existing AWS Identity and Access Management (IAM) user that you created using the IAM interface and you want that user to be able to send emails using the Amazon SES SMTP interface.
- You want to automate SMTP credential generation using code or a script.

For information on each method, see Obtaining Amazon SES SMTP Credentials Using the Amazon SES Console (p. 67) and Obtaining Amazon SES SMTP Credentials by Converting AWS Credentials (p. 68).

Obtaining Amazon SES SMTP Credentials Using the Amazon SES Console

When you generate SMTP credentials by using the Amazon SES console, the Amazon SES console creates an IAM user with the appropriate policies to call Amazon SES and provides you with the SMTP credentials associated with that user.

Note

An IAM user can create Amazon SES SMTP credentials, but the IAM user’s policy must give them permission to use IAM itself, because Amazon SES SMTP credentials are created through IAM. If the IAM user tries to create Amazon SES SMTP credentials using the console and they don’t have IAM permissions, they will get an error that says “… not authorized to perform iam:ListUsers…” In that case, the root account owner needs to modify the IAM user’s policy to allow them to access the following IAM actions: “iam:ListUsers”, “iam:CreateUser”, “iam:CreateAccessKey”, and “iam:PutUserPolicy”.

For a list of SMTP response codes, see SMTP Response Codes Returned by Amazon SES (p. 433).
To create your SMTP credentials

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation pane, choose SMTP Settings.
3. In the content pane, choose Create My SMTP Credentials.
4. In the Create User for SMTP dialog box, you will see that an SMTP user name has been filled in for you. You can accept this suggested user name or enter a different one. To proceed, choose Create.
5. Choose Show User SMTP Credentials. Your SMTP credentials will be displayed on the screen; copy them and store them in a safe place. You can also choose Download Credentials to download a file that contains your credentials.

Important
This is the only time that you will be able to view your SMTP credentials! We strongly advise you to download these credentials and refrain from sharing them with others.

6. Choose Close Window.

If you want to delete your SMTP credentials, go to the IAM console at https://console.aws.amazon.com/iam/ and delete the IAM user name that corresponds with your SMTP credentials. To learn more, go to the Using IAM guide.

If you want to change your SMTP password, go to the IAM console and delete your existing IAM user, and then go to the Amazon SES console to re-generate your SMTP credentials.

Obtaining Amazon SES SMTP Credentials by Converting AWS Credentials

If you have an IAM user that you set up using the IAM interface, you can derive the user’s Amazon SES SMTP credentials from their AWS credentials.

Important
Do not use temporary AWS credentials to derive SMTP credentials. The Amazon SES SMTP interface does not support SMTP credentials that have been generated from temporary security credentials.
To enable the IAM user to send email using the Amazon SES SMTP interface, you need to do the following two steps:

- Derive the user’s SMTP credentials from their AWS credentials using the algorithm provided in this section. Because you are starting from AWS credentials, the SMTP username will be the same as the AWS access key ID, so you just need to generate the SMTP password.

  **Important**
  If you generate SMTP credentials using the Amazon SES console, the SMTP username is not the same as the AWS access key ID. The SMTP username and the AWS access key ID are only the same if you generate the SMTP password programmatically, as described in this section.

- Apply the following policy to the IAM user:

```json
{ "Statement": [ { "Effect":"Allow", "Action":"ses:SendRawEmail", "Resource":"*" } ] }
```

For more information about using Amazon SES with IAM, see Controlling Access to Amazon SES (p. 354).

**Note**
Although you can generate Amazon SES SMTP credentials for any existing IAM user, we recommend for security reasons that you create a separate IAM user for the AWS credentials that you will use to generate the SMTP password. For information about why it is good practice to create users for specific purposes, go to IAM Best Practices.

The following pseudocode shows the algorithm that converts an AWS Secret Access Key to an Amazon SES SMTP password.

```java
key = AWS Secret Access Key;
message = "SendRawEmail";
versionInBytes = 0x02;
signatureInBytes = HmacSha256(message, key);
signatureAndVer = Concatenate(versionInBytes, signatureInBytes);
smtPassword = Base64(signatureAndVer);
```

The following is an example Java implementation that converts an AWS Secret Access Key to an Amazon SES SMTP password. Before you run the program, put the AWS Secret Access Key of the IAM user into an environment variable called AWS_SECRET_ACCESS_KEY. The output of the program is the SMTP password. That password, along with the SMTP username (which, if you generate the SMTP password programmatically, is the same as the AWS access key ID) are the user's Amazon SES SMTP credentials.

```java
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class SesSmtpCredentialGenerator {
  private static final String KEY_ENV_VARIABLE = "AWS_SECRET_ACCESS_KEY"; // Put your AWS secret access key in this environment variable.
  private static final String MESSAGE = "SendRawEmail"; // Used to generate the HMAC signature. Do not modify.
  private static final byte VERSION = 0x02; // Version number. Do not modify.

  public static void main(String[] args) {
    // Get the AWS secret access key from environment variable AWS_SECRET_ACCESS_KEY.
  }
}
```
String key = System.getenv(KEY_ENV_VARIABLE);
if (key == null)
{
    System.out.println("Error: Cannot find environment variable AWS_SECRET_ACCESS_KEY.");
    System.exit(0);
}

// Create an HMAC-SHA256 key from the raw bytes of the AWS secret access key.
SecretKeySpec secretKey = new SecretKeySpec(key.getBytes(), "HmacSHA256");

try {
    // Get an HMAC-SHA256 Mac instance and initialize it with the AWS secret access key.
    Mac mac = Mac.getInstance("HmacSHA256");
    mac.init(secretKey);

    // Compute the HMAC signature on the input data bytes.
    byte[] rawSignature = mac.doFinal(MESSAGE.getBytes());

    // Prepend the version number to the signature.
    byte[] rawSignatureWithVersion = new byte[rawSignature.length + 1];
    byte[] versionArray = {VERSION};
    System.arraycopy(versionArray, 0, rawSignatureWithVersion, 0, 1);
    System.arraycopy(rawSignature, 0, rawSignatureWithVersion, 1, rawSignature.length);

    // To get the final SMTP password, convert the HMAC signature to base 64.
    String smtpPassword = DatatypeConverter.printBase64Binary(rawSignatureWithVersion);
    System.out.println(smtpPassword);
} catch (Exception ex) {
    System.out.println("Error generating SMTP password: " + ex.getMessage());
}
}

### Connecting to the Amazon SES SMTP Endpoint

The following table shows the Amazon SES SMTP endpoints for the regions in which Amazon SES is available.

<table>
<thead>
<tr>
<th>Region name</th>
<th>SMTP endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (N. Virginia)</td>
<td>email-smtp.us-east-1.amazonaws.com</td>
</tr>
<tr>
<td>US West (Oregon)</td>
<td>email-smtp.us-west-2.amazonaws.com</td>
</tr>
<tr>
<td>EU (Ireland)</td>
<td>email-smtp.eu-west-1.amazonaws.com</td>
</tr>
</tbody>
</table>

The Amazon SES SMTP endpoint requires that all connections be encrypted using Transport Layer Security (TLS). (Note that TLS is often referred to by the name of its predecessor protocol, SSL.) Amazon SES supports two mechanisms for establishing a TLS-encrypted connection: STARTTLS and TLS Wrapper. Check the documentation for your software to determine whether it supports STARTTLS, TLS Wrapper, or both.

**Important**

Amazon Elastic Compute Cloud (Amazon EC2) throttles email traffic over port 25 by default. To avoid timeouts when sending email through the SMTP endpoint from EC2, use a different port (587 or 2587) or fill out a Request to Remove Email Sending Limitations to remove the throttle.


**STARTTLS**

STARTTLS is a means of upgrading an unencrypted connection to an encrypted connection. There are versions of STARTTLS for a variety of protocols; the SMTP version is defined in RFC 3207.

To set up a STARTTLS connection, the SMTP client connects to the Amazon SES SMTP endpoint on port 25, 587, or 2587, issues an EHLO command, and waits for the server to announce that it supports the STARTTLS SMTP extension. The client then issues the STARTTLS command, initiating TLS negotiation. When negotiation is complete, the client issues an EHLO command over the new encrypted connection, and the SMTP session proceeds normally.

**TLS Wrapper**

TLS Wrapper (also known as SMTPS or the Handshake Protocol) is a means of initiating an encrypted connection without first establishing an unencrypted connection. With TLS Wrapper, the Amazon SES SMTP endpoint does not perform TLS negotiation: it is the client's responsibility to connect to the endpoint using TLS, and to continue using TLS for the entire conversation. TLS Wrapper is an older protocol, but many clients still support it.

To set up a TLS Wrapper connection, the SMTP client connects to the Amazon SES SMTP endpoint on port 465 or 2465. The server presents its certificate, the client issues an EHLO command, and the SMTP session proceeds normally.

**Configuring Email Clients to Send Through Amazon SES**

After you obtain your SMTP user name and password (p. 67), you can send email through Amazon SES.

The following procedures show how to configure three popular email clients—Mozilla Thunderbird, Microsoft Outlook, and macOS Mail—to send email through Amazon SES. If you are using a different email client, the specific procedures are different, but the concepts and settings are the same.

*Note*

These procedures help you set up an email client to send email using Amazon SES. However, you cannot use these clients to receive email sent to the email addresses you use with Amazon SES. For more information about receiving email in Amazon SES, see Receiving Email (p. 166).

Topics in this section:

- Configuring Mozilla Thunderbird to Send Email Using Amazon SES (p. 71)
- Configuring Microsoft Outlook to Send Email Using Amazon SES (p. 75)
- Configuring macOS Mail to Send Email Using Amazon SES (p. 79)

**Configuring Mozilla Thunderbird to Send Email Using Amazon SES**

The procedures in this section help you configure Mozilla Thunderbird to send email through Amazon SES. These procedures were tested using Mozilla Thunderbird version 52.5 on Windows, macOS and Linux. The procedures may differ slightly for other versions of Thunderbird.

Part 1: Create Local Folders

Amazon SES does not include server-based folders in which you can save items such as sent mail and drafts. For this reason, you must create these folders on your computer. You will configure Thunderbird to save mail to these folders in the section called “Part 3: Configure Thunderbird to Save Sent Mail and Drafts on Your Computer” (p. 73).
To create the Sent Mail and Drafts folders
1. In the bottom left corner of the Thunderbird window, click the **Offline Mode** icon to enable offline mode. If Thunderbird asks if you want to download messages for offline use, choose **Later**.
2. In the navigation pane on the left side of the Thunderbird window, right-click a blank area, and then choose **New Folder**.
3. On the **New Folder** window, complete the following sections:
   - For **Name**, type **Sent Mail**.
   - For **Create as a subfolder of**, choose **Local Folders**.
4. Repeat steps 2 and 3 to create an additional folder, but this time, name the folder **Drafts**.

Part 2: Configure the SMTP Server

Before you can send email through Amazon SES, you must configure Thunderbird to connect to the Amazon SES SMTP endpoint.

**To configure the SMTP server**

1. In Thunderbird, complete one of the following steps:
   - If you use Windows: choose the **Menu** icon, point to **Options**, and then choose **Account Settings**.
   - If you use Linux or macOS: choose the **Menu** icon, point to **Preferences**, and then choose **Account Settings**.
2. On the **Account Settings** window, in the column on the left, choose **Outgoing Server (SMTP)**.
3. Choose **Add**.
4. On the **SMTP Server** window, complete the following sections:
   - For **Description**, type **Amazon SES**.
   - For **Server Name**, enter the SMTP endpoint for the AWS Region in which you use Amazon SES. For a list of endpoints, see the section called “Email Sending Endpoints” (p. 409).
   - For **Port**, type **587**.
   - For **Connection security**, choose **STARTTLS**.
   - For **Authentication method**, choose **Normal password**.
   - For **User Name**, type your SMTP user name.

   **Note**
   Your SMTP user name is not the same as your AWS access key ID. For more information, see **Obtaining Your Amazon SES SMTP Credentials** (p. 67).

   When you finish, choose **OK**.
5. On the **Account Settings** window, choose **Account Actions**, and then choose **Add Mail Account**.
6. On the **Mail Account Setup** window, complete the following sections:
   - For **Your name**, type the name you want to appear on messages sent from this account.
   - For **Email address**, type the email address you use to send email with Amazon SES.
   - Leave the **Password** field blank, and clear the check box next to **Remember password**.
When you finish, choose **Advanced config.** You will return to the **Account Settings** window.

**Note**  
You can only complete this step if Thunderbird is in Offline Mode.

7. On the **Account Settings** window, in the column on the left, choose the account you created in the previous step.

8. For **Outgoing Server (SMTP)**, choose the SMTP server you created in step 8.

### Part 3: Configure Thunderbird to Save Sent Mail and Drafts on Your Computer

In this section, you will find procedures for saving sent mail and drafts to your computer.

**To configure Thunderbird to save sent mail and drafts to your computer**

1. On the **Account Settings** window, under your account, choose **Server Settings**.
2. Under **Server Settings**, clear the check boxes next to the following items:
   - Check for new messages at startup
   - Check for new messages every 10 minutes
   - Allow immediate server notifications when new messages arrive
3. On the **Account Settings** window, under your account, choose **Copies & Folders**.
4. Under **When sending messages, automatically**, choose **Other**, and then choose the Sent Mail folder you created in step 3.

![Copies & Folders](image)

5. Under **Message Archives**, clear the check box next to **Keep message archives in**.

![Message Archives](image)

6. Under **Drafts and Templates**, choose **Other**, and then choose the Drafts folder you created in step 4.
Part 4: Test Email Sending Functionality

Complete the procedures in this section to ensure that Thunderbird is properly configured to send email through Amazon SES.

To test email sending in Thunderbird

1. In the bottom left corner of the screen, choose the Offline mode icon to disable offline mode.
2. On the Thunderbird window, choose Write. Send a test message to yourself, or to another email address that has been verified with Amazon SES.

When you send the email, you may be prompted to enter a password. Enter your Amazon SES SMTP password, and then select the box next to Use Password Manager to remember this password.

Note
Your SMTP password is not the same as your AWS access key ID. For more information, see Obtaining Your Amazon SES SMTP Credentials (p. 67).

The first time you send an email using this configuration, Thunderbird may display a message stating that it was unable to connect to the server. If this message appears, click Retry.

(Optional) Part 5: Specify a Configuration Set When Sending Email

You can configure Thunderbird so that it allows you to specify a configuration set when you send a new message.

Warning
You modify the hidden configuration settings in Thunderbird during this procedure. Changing these settings may render Thunderbird unusable. Proceed with caution.

To add a configuration set header

1. In Thunderbird, choose the Menu icon, point to Options, and then choose Options.
2. On the Options window, choose Advanced. On the General tab, choose Config Editor.
3. On the about:config window, right-click anywhere in the list of configuration settings, point to New, and then choose String.
4. For Enter the preference name, type mail.compose.other.header, and then choose OK.
5. For mail.compose.other.header, type X-SES-CONFIGURATION-SET, and then choose OK.
6. Close the about:config window, and then close the Options window.
7. When you send an email, type the recipient's address in the To field. Click the blank line below the recipient's email address. Click the To menu, and then choose X-SES-CONFIGURATION-SET.
8. On the X-SES-CONFIGURATION-SET line, type the name of the configuration set you want to use. Complete the remaining sections of the email as you normally would.

Configuring Microsoft Outlook to Send Email Using Amazon SES

The procedures in this section will help you configure Microsoft Outlook for Windows to send email through Amazon SES. These procedures were tested using Microsoft Outlook 2016, build 16.0.4549.1000. The procedures may differ slightly for other versions of Outlook.

**Note**
These procedures will not work on Microsoft Outlook for Mac. Outlook for Mac requires an incoming (IMAP) server, and does not allow you to bypass the connection test in the same way that the Windows version does. If you need a graphical client for sending email from a macOS computer, we recommend that you use Thunderbird (p. 71).

To configure Microsoft Outlook 2016 to send email using Amazon SES

1. In Microsoft Outlook, choose File, and then choose Info.
2. Choose Add Account.
3. On the Add Account window, choose Manual setup or additional server types, and then choose Next.
4. Under Choose Service, choose POP or IMAP, and then choose Next.
5. Under POP and IMAP Account Settings, fill in the following fields:
   a. **Your Name** – Your full name.
   b. **Email Address** – The email address from which you will send emails. This address must be verified in Amazon SES, and it must exactly match the address specified in Amazon SES.
   c. **Account Type** – Choose POP3.
   d. **Incoming mail server** – Type none. (Even though you are setting up Amazon SES for outgoing email only, this field is required.)
   e. **Outgoing mail server (SMTP)** – Type the SMTP endpoint for the outgoing mail server. For a list of Amazon SES SMTP endpoints, see Connecting to the Amazon SES SMTP Endpoint (p. 70). For example, if you use the Amazon SES endpoint in the US West (Oregon) Region, the outgoing mail server is email-smtp.us-west-2.amazonaws.com.
   f. **User Name** – Type none. You will configure your credentials later in this procedure.
7. On the Internet E-mail Settings window, choose the Outgoing Server tab, and then complete the following fields:

   a. **My outgoing server (SMTP) requires authentication** – [Selected]
   b. **Log on using** – [Selected]
   c. **User Name** – Your Amazon SES SMTP user name.
      
      **Important**
      Your SMTP password is not the same as your AWS access key ID. For more information, see Obtaining Your Amazon SES SMTP Credentials (p. 67).
   d. **Password** – Your Amazon SES SMTP password.
      
      **Important**
      Your SMTP password is not the same as your AWS secret access key. For more information, see Obtaining Your Amazon SES SMTP Credentials (p. 67).
   e. **Remember Password** – [Selected]
8. Choose the **Advanced** tab, and then complete the following fields:

   a. **Outgoing server (SMTP)** – Type 587.

   b. **Use the following type of encrypted connection** – Choose **TLS**.
9. Choose OK.

10. On the Add Account page, choose Test Account Settings. On the Internet E-mail - none window, choose Cancel.

11. Choose the Tasks tab and confirm that the Send test e-mail message test completed successfully.

   Note
   Because you are using Amazon SES as your outgoing email server only, the Log onto incoming mail server test is expected to fail. The Send test e-mail message test should pass.

12. If the test completes successfully, clear the box next to Automatically test account settings when Next is clicked, and then choose Next.

13. Choose Next, and then choose Finish.

14. You set up Amazon SES for email sending only. To ensure that the account is not set up to receive messages using Amazon SES, you must disable mail retrieval for the account by using the following steps.

   a. In Microsoft Outlook, choose the Send/Receive tab.
   b. On the Send/Receive tab, choose Send/Receive Groups, and then choose Define Send/Receive Groups.
   c. In the Send/Receive Groups dialog box, choose Edit.
   d. In the Accounts section on the left, choose the account you just created for sending mail through Amazon SES.
   e. Under Account Options, clear Receive mail items.
   f. Choose OK, and then choose Close.
Configuring macOS Mail to Send Email Using Amazon SES

The procedures in this section will help you configure the Mail application that is included with macOS to send email through Amazon SES. These procedures were tested using Mail version 11.1 on macOS High Sierra (version 10.13). The procedures may differ slightly for other versions of the Mail app and macOS.

Note
The procedures in this section apply to the version of Mail that is included with macOS desktop and laptop computers. The version of Mail that is included with iOS devices, such as iPhones and iPads, cannot be configured to send mail using Amazon SES.

To configure macOS Mail to send email using Amazon SES

1. In Mail, on the Mailbox menu, choose Take All Accounts Offline.
2. On the Mail menu, choose Add Account.
3. On the Add a Mail Account window, complete the following sections:
   • For Name, type the name that you want to appear on messages sent from this account.
   • For Email Address, type the email address you use to send email with Amazon SES.
   • For Password, type any value. This field is used to connect to an IMAP server. Because Amazon SES does not include an IMAP server, the value you enter here is not important.

   When you finish, choose Sign In. You will see a message stating that Mail is unable to verify your account name or password. Choose Next.
4. On the Select the apps you want to use with this account window, choose Mail, and then choose Done.
5. On the Mail menu, choose Preferences.
6. In the pane on the left, choose the IMAP account you just created.
7. On the Server Settings tab, under Outgoing Mail Server (SMTP), complete the following sections:
   • For User Name, type your SMTP user name.
   • For Password, type your SMTP password.

   Note
   Your SMTP user name and password are not the same as your AWS access key ID and secret access key. For more information, see Obtaining Your Amazon SES SMTP Credentials (p. 67).
   • For Host Name, enter the SMTP endpoint for the AWS Region in which you use Amazon SES. For a list of endpoints, see the section called “Email Sending Endpoints” (p. 409).

   When you finish, choose Save.

   Note
   Do not change any of the settings in the Incoming Mail Server (IMAP) section. If you make changes in this section, you will not be able to save your settings.
8. On the Mailbox menu, choose Take All Accounts Online.
9. Send a test message to another email address you control to test the connection.

Sending Email Through Amazon SES From Software Packages

There are a number of commercial and open source software packages that support sending email via SMTP. Here are some examples:
• Blogging platforms
• RSS aggregators
• List management software
• Workflow systems

You can configure any such SMTP-enabled software to send email through the Amazon SES SMTP interface. For instructions on how to configure SMTP for a particular software package, see the documentation for that software.

The following procedure shows how to set up Amazon SES sending in JIRA, a popular issue-tracking solution. With this configuration, JIRA can notify users via email whenever there is a change in the status of a software issue.

To Configure JIRA to Send Email Using Amazon SES

1. Using your web browser, log in to JIRA with administrator credentials.
2. In the browser window, choose Administration.
3. On the System menu, choose Mail.
4. On the Mail administration page, choose Mail Servers.
5. Choose Configure new SMTP mail server.
6. On the Add SMTP Mail Server form, fill in the following fields:
   a. Name—A descriptive name for this server.
   b. From address—The address from which email will be sent. You will need to verify this email address with Amazon SES before you can send from it. For more information about verification, see Verifying Identities in Amazon SES (p. 43).
   c. Email prefix—A string that JIRA prepends to each subject line prior to sending.
   d. Protocol—Choose SMTP.
      Note
      If you cannot connect to Amazon SES using this setting, try SECURE_SMTP.
   e. Host Name—See Connecting to the Amazon SES SMTP Endpoint (p. 70) for a list of Amazon SES SMTP endpoints. For example, if you want to use the Amazon SES endpoint in the US West (Oregon) region, the host name would be email-smtp.us-west-2.amazonaws.com.
   f. SMTP Port—25, 587, or 2587 (to connect using STARTTLS), or 465 or 2465 (to connect using TLS Wrapper).
   g. TLS—Select this check box.
   h. Username—Your SMTP username.
   i. Password—Your SMTP password.

Settings for TLS Wrapper are shown below.
7. Choose **Test Connection**. If the test email that JIRA sends through Amazon SES arrives successfully, then your configuration is complete.

### Sending Email Through Amazon SES From Your Application

Many programming languages support sending email using SMTP. This capability might be built into the programming language itself, or it might be available as an add-on, plug-in, or library. You can take advantage of this capability by sending email through Amazon SES from within application programs that you write.

For examples in C# and Java, see [Send an Email by Accessing the Amazon SES SMTP Interface Programmatically](p. 19) in the Getting Started section.

### Integrating Amazon SES with Your Existing Email Server

If you currently administer your own email server, you can use the Amazon SES SMTP endpoint to send all of your outgoing email to Amazon SES. There is no need to modify your existing email clients and applications; the changeover to Amazon SES will be transparent to them.

Several mail transfer agents (MTAs) support sending email through SMTP relays. This section provides general guidance on how to configure some popular MTAs to send email using Amazon SES SMTP interface.

- To configure Postfix to send email through Amazon SES, see [Integrating Amazon SES with Postfix](p. 82).
- To configure Sendmail to send email through Amazon SES, see [Integrating Amazon SES with Sendmail](p. 85).
- To configure Microsoft Exchange to send email through Amazon SES, see [Integrating Amazon SES with Microsoft Exchange](p. 87).
- To configure Microsoft Windows Server’s IIS SMTP server to send email through Amazon SES, see [Integrating Amazon SES with Microsoft Windows Server IIS SMTP](p. 93).
To configure Exim to send email through Amazon SES, see Integrating Amazon SES with Exim (p. 94).

The Amazon SES SMTP endpoint requires that all connections be encrypted using Transport Layer Security (TLS).

Integrating Amazon SES with Postfix

Postfix is an alternative to the widely used Sendmail Message Transfer Agent (MTA). For information about Postfix, go to http://www.postfix.org. The procedures in this topic will work with Linux, macOS, or Unix.

Prerequisites

Before you complete the procedures in this section, you must perform the following tasks:

- Uninstall Sendmail, if it is already installed on your system. The procedures for completing this step will vary depending on the operating system you use.
- Install Postfix. The procedures for completing this step will vary depending on the operating system you use.
- Verify an email address or domain that you will use for sending email. If your account is still in the sandbox, you must also verify all of the recipient addresses that you send to. For more information, see Verifying Email Addresses in Amazon SES (p. 43).
- (Optional) If you are sending email through Amazon SES from an Amazon EC2 instance, you may need to assign an Elastic IP Address to your Amazon EC2 instance for the receiving ISP to accept your email. For more information, see Amazon EC2 Elastic IP Addresses.

Configuring Postfix

Complete the following procedures to configure your mail server to send email through Amazon SES using Postfix.

To configure Postfix

1. In a text editor, open the file /etc/postfix/main.cf.
2. Add the following lines to the end of main.cf:

   ```
   relayhost = [email-smtp.us-west-2.amazonaws.com]:587
   smtp_sasl_auth_enable = yes
   smtp_sasl_security_options = noanonymous
   smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
   smtp_use_tls = yes
   smtp_tls_security_level = encrypt
   smtp_tls_note_starttls_offer = yes
   ```

   Note
   If you use Amazon SES in an AWS Region other than US West (Oregon), replace email-smtp.us-west-2.amazonaws.com in the example above with the SMTP endpoint of the appropriate region. For more information, see Regions (p. 409).
3. In a text editor, open the file /etc/postfix/master.cf.
4. In master.cf, locate the following line:

   ```
   -o smtp_fallback_relay=
   ```
Place a # (hash) character at the beginning of the line to comment it out. Save and close master.cf.

5. In a text editor, open the file /etc/postfix/sasl_passwd. If the file does not already exist, create it.

6. Add the following line to sasl_passwd:

```
[email-smtp.us-west-2.amazonaws.com]:587 SMTPUSERNAME:SMTPPASSWORD
```

**Note**
Replace SMTPUSERNAME and SMTPPASSWORD with your SMTP username and password, respectively. Your SMTP user name and password are not the same as your AWS access key ID and secret access key. For more information about credentials, see the section called “Obtaining Your SMTP Credentials” (p. 67).

If you use Amazon SES in an AWS Region other than US West (Oregon), replace email-smtp.us-west-2.amazonaws.com in the example above with the SMTP endpoint of the appropriate region. For more information, see Regions (p. 409).

Save and close sasl_passwd.

7. At a command prompt, type the following command to create a hashmap database file containing your SMTP credentials: `sudo postmap hash:/etc/postfix/sasl_passwd`

8. (Optional) The /etc/postfix/sasl_passwd and /etc/postfix/sasl_passwd.db files you created in the previous steps are not encrypted. Because these files contain your SMTP credentials, we recommend that you modify the files' ownership and permissions in order to restrict access to them. To restrict access to these files:

   a. At a command prompt, type the following command to change the ownership of the files: `sudo chown root:root /etc/postfix/sasl_passwd /etc/postfix/sasl_passwd.db`

   b. At a command prompt, type the following command to change the permissions of the files so that only the root user can read or write to them: `sudo chmod 0600 /etc/postfix/sasl_passwd /etc/postfix/sasl_passwd.db`

9. Tell Postfix where to find the CA certificate (needed to verify the Amazon SES server certificate). The command you use in this step varies based on your operating system.

   - If you use Amazon Linux, Red Hat Enterprise Linux, or a related distribution, type the following command: `sudo postconf -e 'smtp_tls_CAfile = /etc/ssl/certs/ca-bundle.crt'`

   - If you use Ubuntu or a related distribution, type the following command: `sudo postconf -e 'smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt'`

10. Type the following command to start the Postfix server (or to reload the configuration settings if the server is already running): `sudo postfix start; sudo postfix reload`

11. Send a test email by typing the following at a command line, pressing Enter after each line. Replace sender@example.com with your From email address. The From address must be verified for use with Amazon SES. Replace recipient@example.com with the destination address. If your account is still in the sandbox, the recipient address must also be verified. Finally, the final line must contain a single period (.) with no other content.

```
sendmail -f sender@example.com recipient@example.com
From: Sender Name <sender@example.com>
Subject: Amazon SES Test
This message was sent using Amazon SES.
.
83```
Note
In some versions of Ubuntu, when you enter the sendmail -f command, you may see the following error message: command not found: sendmail. In this situation, replace sendmail in the command above with /usr/lib/sendmail.

12. Check the mailbox associated with the recipient address. If the email does not arrive, check your junk mail folder. If you still cannot locate the email, check your system’s mail log (typically located at /var/log/maillog) for more information.

Advanced Usage Example

This example shows how to send an email that uses a configuration set (p. 215), and that uses MIME-multipart encoding to send both a plain text and an HTML version of the message, along with an attachment. It also includes a link tag (p. 453), which can be used for categorizing click events. The content of the email is specified in an external file, so that you do not have to manually type the commands in the Postfix session.

To send a multipart MIME email using Postfix

1. In a text editor, create a new file called mime-email.txt.
2. In the text file, paste the following content, replacing the values in red with the appropriate values for your account:

```plaintext
X-SES-CONFIGURATION-SET:ConfigSet
From:Sender Name <sender@example.com>
Subject:Amazon SES Test
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="YWVhZDF1Y2QzMGQ2N2U0YTZmODU"

--YWVhZDF1Y2QzMGQ2N2U0YTZmODU
Content-Type: multipart/alternative; boundary="3NjM0N2QwMTE4MWQ0ZTg2NTYxZQ"

--3NjM0N2QwMTE4MWQ0ZTg2NTYxZQ
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: quoted-printable
Amazon SES Test
This message was sent from Amazon SES using the SMTP interface.
For more information, see:

--3NjM0N2QwMTE4MWQ0ZTg2NTYxZQ
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: quoted-printable
<html>
<head>
</head>
<body>
<h1>Amazon SES Test</h1>
<p>This message was sent from Amazon SES using the SMTP interface.</p>
<p>For more information, see
Using the Amazon SES SMTP Interface to Send Email</a> in the Amazon SES Developer Guide.</p>
</body>
</html>
```

84
Save and close the file.

3. At the command line, type the following command: `sendmail recipient@example.com < mime-email.txt`

   If the command runs successfully, it exits without providing any output.

4. Check your inbox for the email. If the message was not delivered, check your system's mail log.

Integrating Amazon SES with Sendmail

Sendmail was released in the early 1980s, and has been continuously improved ever since. It is a very flexible and configurable MTA, and it has a large installed base. For information about Sendmail, go to [http://www.sendmail.com/sm/open_source/](http://www.sendmail.com/sm/open_source/).

The following instructions show you how to configure Sendmail to send email through Amazon SES.

These instructions were tested on a 64-bit Amazon EC2 instance using the following Amazon Machine Image (AMI):

- Amazon Linux AMI 2015.09.2 (ami-8fcee4e5)

To launch an Amazon EC2 instance, which includes selecting an AMI, see [Amazon Machine Images (AMIs)](https://aws.amazon.com/marketplace/pp/B01N8I29T6).

Prerequisites

Before you perform one of the following procedures, verify the following:

- The Sendmail package is installed on your computer, and you are able to successfully send an email using Sendmail without Amazon SES.

  **Note**

  To see if a package is installed on a computer running Red Hat Linux, type `rpm -qa | grep <package>`, where `<package>` is the package name. To see if a package is installed on a computer running Ubuntu Linux, type `dpkg -s <package>`.

- In addition to the Sendmail package, the following packages are installed on your computer: sendmail-cf, m4, and cyrus-sasl-plain.

- You have verified your “From” address and, if your account is still in the sandbox, you have also verified your “To” addresses. For more information, see [Verifying Email Addresses in Amazon SES](#).

- (Optional) If you are sending email through Amazon SES from an Amazon EC2 instance, you may need to assign an Elastic IP Address to your Amazon EC2 instance for the receiving ISP to accept your email. For more information, see [Amazon EC2 Elastic IP Addresses](#).

- (Optional) If you are sending email through Amazon SES from an Amazon EC2 instance, you can fill out a Request to Remove Email Sending Limitations to remove the additional sending limit restrictions that are applied to port 25 by default.
To configure Sendmail to send email through the Amazon SES endpoint in US West (Oregon) using STARTTLS

1. Open the /etc/mail/authinfo file for editing. If the file does not exist, create it.

   Important
   These instructions assume that you want to use Amazon SES in the US West (Oregon) AWS region. If you want to use a different region, replace all instances of email-smtp.us-west-2.amazonaws.com in these instructions with the SMTP endpoint of the desired region. For a list of SMTP endpoints, see Regions and Amazon SES (p. 409).

2. Add the following line to /etc/mail/authinfo, where:

   • U:root—Do not modify.
   • I:USERNAME—Replace USERNAME with the Amazon SES username you obtained using the instructions in Obtaining Your Amazon SES SMTP Credentials (p. 67). This is NOT the same as your AWS Access Key ID.
   • P:PASSWORD—Replace PASSWORD with the Amazon SES password you obtained using the instructions in Obtaining Your Amazon SES SMTP Credentials (p. 67). This is NOT the same as your AWS Secret Key.
   • M:LOGIN—Replace LOGIN with the method of authentication to use. For example, PLAIN, DIGEST-MD5, etc.

   AuthInfo:email-smtp.us-west-2.amazonaws.com "U:root" "I:USERNAME" "P:PASSWORD" "M:LOGIN"

   If Sendmail cannot authenticate with the Amazon SES SMTP endpoint because the hostname does not match, try adding the additional line specified in Amazon SES SMTP Issues (p. 430).

3. Save the authinfo file.

4. At a command prompt, type the following command to generate /etc/mail/authinfo.db:

   sudo makemap hash /etc/mail/authinfo.db < /etc/mail/authinfo

5. Open the /etc/mail/access file and include support for relaying to the Amazon SES SMTP endpoint by adding the following line. If Sendmail cannot authenticate with the Amazon SES SMTP endpoint because the hostname does not match, try adding the additional line specified in Amazon SES SMTP Issues (p. 430).

   Connect:email-smtp.us-west-2.amazonaws.com RELAY

   Save the file.

6. At a command prompt, type the following command to regenerate /etc/mail/access.db:

   sudo makemap hash /etc/mail/access.db < /etc/mail/access

7. Save a back-up copy of /etc/mail/sendmail.mc and /etc/mail/sendmail.cf.

8. Add the following group of lines to the /etc/mail/sendmail.mc file before any MAILER() definitions. If you add a FEATURE() line after a MAILER() definition, when you run m4 in a subsequent step, you will get the following error: "ERROR: FEATURE() should be before MAILER().":

   Important
   If you are using an AWS region other than US West (Oregon), replace the SMART_HOST value with the Amazon SES SMTP endpoint of the region you're using, and be sure to use the ` character and the apostrophe exactly as shown.

   define(`SMART_HOST', `email-smtp.us-west-2.amazonaws.com')dnl
In the text you just added to `sendmail.mc`, in the line that starts with `MASQUERADE_AS`, replace `YOUR_DOMAIN` with the domain name from which you are sending your email. By adding this masquerade, you are making email from this host appear to be sent from your domain. Otherwise, the email will appear as if the email is being sent from the host name of the mail server, and you may get an "Email address not verified" error when you try to send an email.

10. Save the `sendmail.mc` file.

11. At a command prompt, type the following command to make `sendmail.cf` writeable:

   ```
sudo chmod 666 /etc/mail/sendmail.cf
   
   ```

12. At a command prompt, type the following command to regenerate `sendmail.cf`:

   ```
sudo m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
   
   ```

   **Note**
   
   If you encounter errors such as "Command not found" and "No such file or directory," make sure you have installed the m4 and sendmail-cf packages as specified in the prerequisites section above.

13. At a command prompt, type the following command to reset the permissions of `sendmail.cf` to read only:

   ```
sudo chmod 644 /etc/mail/sendmail.cf
   
   ```

14. At a command prompt, type the following command to restart Sendmail:

   ```
sudo /etc/init.d/sendmail restart
   
   ```

15. Send a test email by doing the following:

   a. At a command prompt, type the following. Note that you should replace `from@example.com` with your "From" email address, which you must have verified with Amazon SES. Replace `to@example.com` with your "To" address. If your account is still in the sandbox, the "To" address must also be verified.

   ```
sudo /usr/sbin/sendmail -f from@example.com to@example.com
   
   ```

   b. Press <Enter>. Type the body of the message, pressing <Enter> after each line.

   c. When you are finished typing the email, press CTRL+D to send the email.

16. Check the recipient email's client for the email. If you cannot find the email, check the Junk box in the recipient's email client. If you still cannot find the email, look at the Sendmail log on the mail server. The log is typically in `/var/spool/mail/<user>`.

### Integrating Amazon SES with Microsoft Exchange

You can configure Microsoft Exchange to send email through Amazon SES. The following procedures show you how to integrate Microsoft Exchange with Amazon SES using the Microsoft Exchange GUI or Windows PowerShell.

**Important**

Follow only one of the following procedures (Microsoft Exchange GUI or Windows PowerShell). If you follow both procedures, you will get an error stating that you have two send connectors with the same name.
These instructions were written using Microsoft Exchange 2013.

**To integrate Microsoft Exchange with Amazon SES using the Microsoft Exchange GUI**

1. Go to the Microsoft Exchange admin center (typically https://<CASServerName>/ecp) and sign in as a user who is part of the Exchange administrators group.
2. From the left menu, choose **mail flow**.

3. Choose **send connectors**.

4. Choose the plus sign.

5. Enter a name for the send connector (for example, SES).

6. Under **Type**, select **Internet**.
7. Choose **Next**.
8. Select **Route mail through smart hosts**.

**New send connector**

Create a Send connector.

There are four types of send connectors. Each connector has different permissions and network settings. [Learn more...](#)

**Name:**

| SES |

**Type:**

- Custom (For example, to send mail to other non-Exchange servers)
- Internal (For example, to send intranet mail)
- Internet (For example, to send internet mail)
- Partner (For example, to route mail to trusted third-party servers)

**Network settings:**

Specify how to send mail with this connector.

- MIX record associated with recipient domain
- Route mail through smart hosts

- Use the external DNS lookup settings on servers with transport roles
9. Choose the plus sign and then enter the Amazon SES endpoint that you will use (for example, email-smtp.us-west-2.amazonaws.com). For a list of Amazon SES endpoints, see Regions and Amazon SES (p. 409).

10. Choose Save. The endpoint you entered will appear in the SMART HOST box.

11. Choose Next.

12. Select Basic authentication, then select Offer basic authentication only after starting TLS, and then enter your Amazon SES SMTP user name and password.

   **Important**
   Your SMTP user name and password are not the same as your AWS access key ID and secret access key. Do not attempt to use your AWS credentials to authenticate yourself against the SMTP endpoint. For more information about credentials, see Using Credentials With Amazon SES (p. 369).

13. Choose Next.

14. Choose the plus sign.

15. Verify that **Type** is SMTP, **FQDN** is *, and **Cost** is 1.
16. Choose **Save** and then choose **Next**.

17. Choose the plus sign.

18. Select all transport servers you would like to apply this rule to and choose **Add**. When you have added all the servers you want to send email through Amazon SES, choose **ok**.

19. Verify that the servers are added and then choose **finish**.
You should now see a send connector for Amazon SES with an enabled status. All outbound mail will now flow through Amazon SES.

To integrate Microsoft Exchange with Amazon SES using Windows PowerShell


2. In the dialog box, enter your Amazon SES SMTP user name and password and then choose OK.

   **Important**
   Your SMTP user name and password are not the same as your AWS access key ID and secret access key. Do not attempt to use your AWS credentials to authenticate yourself against the SMTP endpoint. For more information about credentials, see Using Credentials With Amazon SES (p. 369).
3. At the command prompt, type the following line, replacing ENDPOINT with an Amazon SES SMTP endpoint (for example, email-smtp.us-west-2.amazonaws.com). For a list of Amazon SES endpoints, see Regions and Amazon SES (p. 409).

```powershell
New-SendConnector -Name "SES" -AddressSpaces "+;1" -SmartHosts "ENDPOINT" -SmartHostAuthMechanism BasicAuthRequireTLS -Usage Internet -AuthenticationCredential $creds
```

The command line should now display a send connector for Amazon SES with an enabled status. All outbound mail will now flow through Amazon SES.

### Integrating Amazon SES with Microsoft Windows Server IIS SMTP

You can configure Microsoft Windows Server’s IIS SMTP server to send email through Amazon SES. These instructions were written using Microsoft Windows Server 2012 on an Amazon EC2 instance. You can use the same configuration on Microsoft Windows Server 2008 and Microsoft Windows Server 2008 R2.

**To integrate the Microsoft Windows Server IIS SMTP server with Amazon SES**

1. First, set up Microsoft Windows Server 2012 using the following instructions.
   a. From the Amazon EC2 management console, launch a new Microsoft Windows Server 2012 Base Amazon EC2 instance.
   b. Connect to the instance and log into it using Remote Desktop by following the instructions in Getting Started with Amazon EC2 Windows Instances.
   c. Launch the Server Manager Dashboard.
   d. Install the **Web Server** role. Be sure to include the **IIS 6 Management Compatibility tools** (an option under the **Web Server** checkbox).
   e. Install the **SMTP Server** feature.
2. Next, configure the IIS SMTP service using the following instructions.
   a. Return to the Server Manager Dashboard.
   b. From the **Tools** menu, choose **Internet Information Services (IIS) 6.0 Manager**.
   c. Right-click **SMTP Virtual Server #1** and then select **Properties**.
   d. On the **Access** tab, under **Relay Restrictions**, choose **Relay**.
   e. In the **Relay Restrictions** dialog box, choose **Add**.
   f. Under **Single Computer**, enter 127.0.0.1 for the IP address. You have now granted access for this server to relay email to Amazon SES through the IIS SMTP service.

   In this procedure, we assume that your emails are generated on this server. If the application that generates the email runs on a separate server, you need to grant relaying access for that server in IIS SMTP.

   **Note**
   To extend the SMTP relay to private subnets, for **Relay Restriction**, use **Single Computer** 127.0.0.1 and **Group of Computers** 172.1.1.0 - 255.255.255.0 (in the netmask section). For **Connection**, use **Single Computer** 127.0.0.1 and **Group of Computers** 172.1.1.0 - 255.255.255.0 (in the netmask section).

3. Finally, configure the server to send email through Amazon SES using the following instructions.
a. Return to the **SMTP Virtual Server #1 Properties** dialog box and then choose the **Delivery** tab.
b. On the **Delivery** tab, choose **Outbound Security**.
c. Select **Basic Authentication** and then enter your Amazon SES SMTP username and password. You can obtain these credentials from the Amazon SES console using the procedure in **Obtaining Your Amazon SES SMTP Credentials**.

   **Important**
   Your SMTP user name and password are not the same as your AWS access key ID and secret access key. Do not attempt to use your AWS credentials to authenticate yourself against the SMTP endpoint. For more information about credentials, see **Using Credentials With Amazon SES**.

d. Ensure that **TLS encryption** is selected.
e. Return to the **Delivery** tab.
f. Choose **Outbound Connections**.
g. In the **Outbound Connections** dialog box, ensure that the port is 25 or 587.
h. Choose **Advanced**.
i. For the **Smart host name**, enter the Amazon SES endpoint that you will use (for example, `email-smtp.us-west-2.amazonaws.com`). For a list of Amazon SES endpoints, see **Regions and Amazon SES**.

j. Return to the Server Manager Dashboard.
k. On the Server Manager Dashboard, right-click **SMTP Virtual Server #1** and then restart the service to pick up the new configuration.
l. Send an email through this server. You can examine the message headers to confirm that it was delivered through Amazon SES.

### Integrating Amazon SES with Exim

Exim is an MTA that was originally developed for Unix-like systems. It is a general purpose mail program that is very flexible and configurable.

To learn more about Exim, go to [http://www.exim.org](http://www.exim.org).

**To configure integration with the Amazon SES US West (Oregon) endpoint using STARTTLS**

1. Open the `/etc/exim/exim.conf` file for editing. If the file does not exist, create it.

   **Important**
   These instructions assume that you want to use Amazon SES in the US West (Oregon) AWS Region. If you want to use a different region, replace all instances of `email-smtp.us-west-2.amazonaws.com` in these instructions with the SMTP endpoint of the desired region.
   For a list of SMTP endpoints, see **Regions and Amazon SES**.

2. In `/etc/exim/exim.conf`, make the following changes:

   a. In the **routers** section, after the `begin routers` line, add the following:

   ```
   send_via_ses:
   driver = manualroute
   domains = ! +local_domains
   transport = ses_smtp
   route_list = * email-smtp.us-west-2.amazonaws.com;
   ```

   b. In the **transports** section, after the `begin transports` line, add the following:

   ```
   ses_smtp:
   ```
Using the SMTP Interface

c. In the **authenticators** section, after the **begin authenticators** line, add the following, replacing **USERNAME** and **PASSWORD** with your SMTP user name and password:

```plaintext
Important
Use your SMTP user name and password, not your AWS access key ID and secret access key. Your SMTP credentials and your AWS credentials are not the same. For information about how to obtain your SMTP credentials, see obtaining your Amazon SES SMTP Credentials (p. 67).

ses_login:
driver = plaintext
public_name = LOGIN
client_send = : USERNAME : PASSWORD
```

3. Save the `/etc/exim/exim.conf` file.

When you have finished updating the configuration, restart Exim. At the command line, type the following command and press ENTER.

```
sudo /etc/init.d/exim restart
```

**Note**
This command may not be exactly the same on your particular server.

When you have completed this procedure, your outgoing email will be sent via Amazon SES. To test your configuration, send an email message through your Exim server, and then verify that arrives at its destination. If the message is not delivered, then check your system's mail log for errors. On many systems, this is the `/var/log/exim/main.log` file.

Using the Command Line to Send Email Through the Amazon SES SMTP Interface

You can interact with the Amazon SES SMTP interface from the command line by using a common utility called OpenSSL. In most cases, you'll only use OpenSSL to test your ability to connect to the SMTP interface. However, you can also use OpenSSL to write your own applications that send email using Amazon SES.

OpenSSL is an open-source utility, and is compatible with all operating systems. It is included by default in most versions of Linux and macOS, and can be compiled to run on Windows. To learn more about OpenSSL, see [https://www.openssl.org](https://www.openssl.org).

**Prerequisites**

In order to connect to the Amazon SES SMTP interface using OpenSSL, you must first obtain your SMTP credentials.

**Important**
Your SMTP credentials are different from your standard AWS credentials. For more information about obtaining your SMTP credentials, see the section called “Obtaining Your SMTP Credentials” (p. 67).

After you obtain your SMTP credentials, you must encode them using base64 encoding.
To encode your SMTP user name and password using base64

1. At the command line, type the following command to encode your SMTP user name, replacing \texttt{SMTPUsername} with your SMTP user name:

\begin{verbatim}
echo -n "SMTPUsername" | openssl enc -base64
\end{verbatim}

Make a note of the output of this command.

2. At the command line, type the following command to encode your SMTP password, replacing \texttt{SMTPPassword} with your SMTP password:

\begin{verbatim}
echo -n "SMTPPassword" | openssl enc -base64
\end{verbatim}

Make a note of the output of this command.

Testing Your Connection to the Amazon SES SMTP Interface

There are two ways to connect to the Amazon SES SMTP interface with OpenSSL: using STARTTLS on port 587, and using SSL on port 465. Both options offer the same capabilities. If you are unsure of which option to choose, we recommend using STARTTLS on port 587.

STARTTLS (Port 587)

To connect to the SMTP interface using STARTTLS

• At the command line, type the following command to connect to the Amazon SES SMTP server:

\begin{verbatim}
openssl s_client -crlf -quiet -starttls smtp -connect email-smtp.us-west-2.amazonaws.com:587
\end{verbatim}

\textbf{Note}

Replace \texttt{email-smtp.us-west-2.amazonaws.com} with the URL of the Amazon SES SMTP endpoint for your AWS Region. For more information, see Regions (p. 409).

If the connection was successful, you will see output similar to the following:

\begin{verbatim}
depth=1 C = US, O = Symantec Corporation, OU = Symantec Trust Network, CN = Symantec Class 3 Secure Server CA - G4
250 Ok
\end{verbatim}

The connection automatically closes after about 10 seconds of inactivity.

SSL (Port 465)

To connect to the SMTP interface using SSL

• At the command line, type the following command to connect to the Amazon SES SMTP server:

\begin{verbatim}
openssl s_client -crlf -quiet -connect email-smtp.us-west-2.amazonaws.com:465
\end{verbatim}

\textbf{Note}

Replace \texttt{email-smtp.us-west-2.amazonaws.com} with the URL of the Amazon SES SMTP endpoint for your AWS Region. For more information, see Regions (p. 409).
Using OpenSSl to Send Email Using the Amazon SES SMTP Interface

When an email sender (a client) connects to an SMTP server, the client issues a standard set of requests, and the server replies to each request with a standard response. This series of requests and responses is called an SMTP conversation. When you connect to the Amazon SES SMTP server using OpenSSl, the server expects an SMTP conversation to occur.

In this example, you'll add all of the client requests to a text file, and then use that file as input to one of the OpenSSl commands listed in the previous section.

STARTTLS (Port 587)

To send an email from the command line using the SMTP interface

1. In a text editor, create a new file. Paste the following code into the file.

   ```
   EHLO example.com
   AUTH LOGIN
   Base64EncodedSMTPUserName
   Base64EncodedSMTPPassword
   MAIL FROM: sender@example.com
   RCPT TO: recipient@example.com
   DATA
   X-SES-CONFIGURATION-SET: ConfigSet
   From: Sender Name <sender@example.com>
   To: recipient@example.com
   Subject: Amazon SES SMTP Test
   This message was sent using the Amazon SES SMTP interface.
   .
   QUIT
   ```

2. Make the following changes to the file you created in the previous step:

   - Replace `example.com` with your sending domain.
   - Replace `Base64EncodedSMTPUserName` with your base64-encoded SMTP user name.
   - Replace `Base64EncodedSMTPPassword` with your base64-encoded SMTP password.
   - Replace `sender@example.com` with the email address you are sending from. This identity must be verified.
   - Replace `recipient@example.com` with the destination email address. If your Amazon SES account is still in the sandbox, this address must be verified.
   - Replace `ConfigSet` with the name of the configuration set that you want to apply to this email.

   **Note**
   If you do not want to use a configuration set, remove the entire line that begins with "X-SES-CONFIGURATION-SET".

3. Save the file as `input.txt`.
4. At the command line, type the following command:

```
openssl s_client -crlf -quiet -starttls smtp -connect email-smtp.us-west-2.amazonaws.com:587 < input.txt
```

**Note**
Replace `email-smtp.us-west-2.amazonaws.com` with the URL of the Amazon SES SMTP endpoint for your AWS Region. For more information, see Regions (p. 409).

If the message was sent, you'll see output similar to the following:

```
250 Ok 01010160d7de98d8-21e57d9a-JZho-416c-bbe1-8ebaAexample-000000
```

The string of numbers and text that follows `250 Ok` is the message ID of the email.

**Note**
The connection may remain open after the message is sent. If it does, it will automatically close after about 10 seconds of inactivity.

### SSL (Port 465)

**To send an email from the command line using the SMTP interface**

1. In a text editor, create a new file. Paste the following code into the file.

```
EHLO example.com
AUTH LOGIN
Base64EncodedSMTPUserName
Base64EncodedSMTPPassword
MAIL FROM: sender@example.com
RCPT TO: recipient@example.com
DATA
X-SES-CONFIGURATION-SET: ConfigSet
From: Sender Name <sender@example.com>
To: recipient@example.com
Subject: Amazon SES SMTP Test

This message was sent using the Amazon SES SMTP interface.
.
QUIT
```

2. Make the following changes to the file you created in the previous step:

- Replace `example.com` with your sending domain.
- Replace `Base64EncodedSMTPUserName` with your base64-encoded SMTP user name.
- Replace `Base64EncodedSMTPPassword` with your base64-encoded SMTP password.
- Replace `sender@example.com` with the email address you are sending from. This identity must be verified.
- Replace `recipient@example.com` with the destination email address. If your Amazon SES account is still in the sandbox, this address must be verified.
- Replace `ConfigSet` with the name of the configuration set that you want to apply to this email.

**Note**
If you do not want to use a configuration set, remove the entire line that begins with "X-SES-CONFIGURATION-SET".

3. Save the file as `input.txt`.
4. At the command line, type the following command:

```
openssl s_client -crlf -quiet -connect email-smtp.us-west-2.amazonaws.com:465 < input.txt
```

**Note**
Replace `email-smtp.us-west-2.amazonaws.com` with the URL of the Amazon SES SMTP endpoint for your AWS Region. For more information, see Regions (p. 409).

If the message was sent, you'll see output similar to the following:

```
250 Ok 01010160d7de98d8-21e57d9a-32ho-416c-bbe1-8ebaAexample-000000
```

The string of numbers and text that follows 250 Ok is the message ID of the email.

**Note**
The connection may remain open after the message is sent. If it does, it will automatically close after about 10 seconds of inactivity.

## Using the Amazon SES API to Send Email

To send production email through Amazon SES, you can use the Simple Mail Transfer Protocol (SMTP) interface or the Amazon SES API. For more information about the SMTP interface, see Using the Amazon SES SMTP Interface to Send Email (p. 66). This section describes how to send email by using the API.

The Amazon SES API has a Query interface over HTTPS. See Regions and Amazon SES (p. 409) for a list of Amazon SES API endpoints. When you send an email by using the API, you can provide limited information and have Amazon SES assemble the email for you, or you can assemble the email yourself so that you have complete control over the content and formatting. For more information about the API, see the Amazon Simple Email Service API Reference. You can call the API in the following four ways:

- **Make raw Query requests and responses**—This is the most advanced method, because you are calling the API directly. For information about how to make Query requests and responses, see Amazon SES Query API (p. 371).
- **Use an AWS SDK**—AWS SDKs wrap the low-level functionality of the Amazon SES API with higher-level data types and function calls that take care of the details for you, and provide basic functions (not included in the Amazon SES API), such as request authentication, request retries, and error handling.
- **Use a command line interface**—The AWS Command Line Interface is the command line tool for Amazon SES. We also offer the AWS Tools for Windows PowerShell for those who script in the PowerShell environment.

Regardless of whether you access the Amazon SES API directly or indirectly through an AWS SDK, the AWS Command Line Interface or the AWS Tools for Windows PowerShell, the Amazon SES API provides two different ways for you to send an email, depending on how much control you want over the composition of the email message:

- **Formatted**—Amazon SES composes and sends a properly formatted email message. You need only supply "From:" and "To:" addresses, a subject, and a message body. Amazon SES takes care of all the rest. For more information, see Sending Formatted Email Using the Amazon SES API (p. 100).
- **Raw**—You manually compose and send an email message, specifying your own email headers and MIME types. If you are experienced in formatting your own email, the raw interface gives you more control over the composition of your message. For more information, see Sending Raw Email Using the Amazon SES API (p. 100).
Sending Formatted Email Using the Amazon SES API

You can send a formatted email by using the AWS Management Console or by calling the Amazon SES API through an application directly, or indirectly through an AWS SDK, the AWS Command Line Interface, or the AWS Tools for Windows PowerShell.

The Amazon SES API provides the `SendEmail` action, which lets you compose and send a formatted email. `SendEmail` requires a `From:` address, `To:` address, message subject, and message body—text, HTML, or both. For a complete description of `SendEmail`, go to the Amazon Simple Email Service API Reference.

**Note**

The email address string must be 7-bit ASCII. If you want to send to or from email addresses that contain unicode characters in the domain part of an address, you must encode the domain using Punycode. For more information, see RFC 3492.

For an example of how to compose a formatted message using the AWS SDK for Java or the AWS SDK for .NET, see Send an Email Using the AWS SDK for Java (p. 32) or Send an Email Using the AWS SDK for .NET (p. 29), respectively.

For tips on how to increase your email sending speed when you make multiple calls to `SendEmail`, see Increasing Throughput with Amazon SES (p. 429).

Sending Raw Email Using the Amazon SES API

You can use the Amazon SES `SendRawEmail` operation to send highly customized messages to your recipients.

This section includes procedures for constructing and sending raw email using the Amazon SES API.

About Email Header Fields

Simple Mail Transfer Protocol (SMTP) specifies how email messages are to be sent by defining the mail envelope and some of its parameters, but it does not concern itself with the content of the message. Instead, the Internet Message Format (RFC 5322) defines how the message is to be constructed.

With the Internet Message Format specification, every email message consists of a header and a body. The header consists of message metadata, and the body contains the message itself. For more information about email headers and bodies, see Email Format and Amazon SES (p. 14).

Using MIME

The SMTP protocol is designed for sending email messages composed of 7-bit ASCII characters. While this works well for many use cases, it is insufficient for non-ASCII text encodings (such as Unicode), binary content, or attachments. The Multipurpose Internet Mail Extensions standard (MIME) was developed to overcome these limitations, making it possible to send many other kinds of content using SMTP.

The MIME standard works by breaking the message body into multiple parts and then specifying what is to be done with each part. For example, one part of an email message body might be plain text, while another might be HTML. In addition, MIME allows email messages to contain one or more attachments. Message recipients can view the attachments from within their email clients, or they can save the attachments.

The message header and content are separated by a blank line. Each part of the email is separated by a boundary, a string of characters that denotes the beginning and ending of each part.

The multipart message in the following example contains a text and an HTML part. It also contains an attachment.

```
From: "Sender Name" <sender@example.com>
```
The content type for the message is multipart/mixed, which indicates that the message has many parts (in this example, a body and an attachment), and the receiving client must handle each part separately. Nested within the body section, there is a second part that uses the multipart/alternative content type. This content type indicates that each part contains alternative versions of the same content (in this case, a text version and an HTML version). When sent, if the recipient's email client can display HTML, then the HTML version of the body text is displayed. If the recipient's email client cannot display HTML, then the plain text version is shown. Both versions of the message will also contain an attachment (in this case, a short text file that contains some customer names).

When you nest a MIME part within another part, as in this example, the nested part must use a boundary parameter that is distinct from the boundary parameter in the parent part. These boundaries should be unique strings of characters. To define a boundary between MIME parts, type two hyphens (--) followed by the boundary string. At the end of a MIME part, place two hyphens at both the beginning and the end of the boundary string.

**MIME Encoding**

Because of the 7-bit ASCII restriction of SMTP, any content containing 8-bit characters must first be converted to 7-bit ASCII before sending. MIME defines a `Content-Transfer-Encoding` header field for this purpose.
By convention, the most common encoding scheme is base64, where 8-bit binary content is encoded using a well-defined set of 7-bit ASCII characters. Upon receipt, the email client inspects the Content-Transfer-Encoding header field, and can immediately perform a base64 decode operation on the content, thus returning it to its original form. With most email clients, the encoding and decoding occur automatically, and the user need not be aware of it.

In the example above, the "customers.txt" attachment must be decoded from base64 format in order to be read. Some email clients will encode all MIME parts in base64 format, even if they were originally in plain text. This is not normally an issue, since email clients perform the encoding and decoding automatically.

Note
For a list of MIME types that Amazon SES accepts, see Appendix: Unsupported Attachment Types (p. 456).

If you want certain parts of a message, like some headers, to contain characters other than 7-bit ASCII, then you must use MIME encoded-word syntax (RFC 2047) instead of a literal string. MIME encoded-word syntax uses the following form: =?charset?encoding?encoded-text?. For more information, see RFC 2047. If you want to send to or from email addresses that contain unicode characters in the domain part of an address, you must encode the domain using Punycode. For more information, see RFC 3492.

Sending a Raw Email Using the Amazon SES API

The Amazon SES API provides the SendRawEmail action, which lets you compose and send an email message in the format that you specify. For a complete description of SendRawEmail, see the Amazon Simple Email Service API Reference.

Note
For tips on how to increase your email sending speed when you make multiple calls to SendRawEmail, see Increasing Throughput with Amazon SES (p. 429).

The message body must contain a properly formatted, raw email message, with appropriate header fields and message body encoding. Although it is possible to construct the raw message manually within an application, it is much easier to do so using existing mail libraries.

Java

The following code example shows how to use the JavaMail library and the AWS SDK for Java to compose and send a raw email.

```java
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.PrintStream;
import java.nio.ByteBuffer;
import java.util.Properties;

// JavaMail libraries. Download the JavaMail API from
// https://javaeae.github.io/javamail. After you download the jar file,
// include it in the build path for your project.
import javax.activation.DataHandler;
import javax.activation.DataSource;
import javax.activation.FileDataSource;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.Session;
import javax.mail.internet.AddressException;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeBodyPart;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.MimeUtility;

// AWS SDK libraries. Download the AWS SDK for Java
```
public class AmazonSESSample {
    // Replace sender@example.com with your "From" address.
    // This address must be verified with Amazon SES.
    private static String SENDER = "Sender Name <sender@example.com>");

    // Replace recipient@example.com with a "To" address. If your account
    // is still in the sandbox, this address must be verified.
    private static String RECIPIENT = "recipient@example.com";

    // Specify a configuration set. If you do not want to use a configuration
    // set, comment the following variable, and the
    // ConfigurationSetName=CONFIGURATION_SET argument below.
    private static String CONFIGURATION_SET = "ConfigSet";

    // The subject line for the email.
    private static String SUBJECT = "Customer service contact info";

    // The full path to the file that will be attached to the email.
    // If you are using Windows, escape backslashes as shown in this variable.
    private static String ATTACHMENT = "C:\Users\sender\customers-to-contact.xlsx";

    // The email body for recipients with non-HTML email clients.
    private static String BODY_TEXT = "Hello,
    + "Please see the attached file for a list "
    + "of customers to contact."

    // The HTML body of the email.
    private static String BODY_HTML = "&lt;html&gt;
    + &lt;head&gt;&lt;/head&gt;
    + &lt;body&gt;
    + &lt;h1&gt;Hello!&lt;/h1&gt;
    + &lt;p&gt;Please see the attached file for a 
    + "list of customers to contact.&lt;/p&gt;
    + &lt;/body&gt;
    + &lt;/html&gt;"

    public static void main(String[] args) throws AddressException, MessagingException, IOException {
        String DefaultCharSet = MimeUtility.getDefaultJavaCharset();
        Session session = Session.getDefaultInstance(new Properties());

        // Create a new MimeMessage object.
        MimeMessage message = new MimeMessage(session);

        // Add subject, from and to lines.
        message.setSubject(SUBJECT, "UTF-8");
        message.setFrom(new InternetAddress(SENDER));
        message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(RECIPIENT));

        // Create a multipart/alternative child container.
        MimeMultipart msg_body = new MimeMultipart("alternative");

        // Create a wrapper for the HTML and text parts.
        MimeBodyPart wrap = new MimeBodyPart();
// Define the text part.
MimeBodyPart textPart = new MimeBodyPart();
// Encode the text content and set the character encoding. This step is
// necessary if you're sending a message with characters outside the
// ASCII range.
textPart.set_content(MimeUtility
 .encodeText(BODY_TEXT, DefaultCharSet, "B"), "text/plain;
 charset=UTF-8");
textPart.setHeader("Content-Transfer-Encoding", "base64");

// Define the HTML part.
MimeBodyPart htmlPart = new MimeBodyPart();
// Encode the HTML content and set the character encoding.
htmlPart.set_content(MimeUtility
 .encodeText(BODY_HTML, DefaultCharSet, "B"), "text/html;
 charset=UTF-8");
htmlPart.setHeader("Content-Transfer-Encoding", "base64");

// Add the text and HTML parts to the child container.
msg_body.addBodyPart(textPart);
msg_body.addBodyPart(htmlPart);

// Add the child container to the wrapper object.
wrap.setContent(msg_body);

// Create a multipart/mixed parent container.
MimeMultipart msg = new MimeMultipart("mixed");

// Add the parent container to the message.
message.setContent(msg);

// Add the multipart/alternative part to the message.
msg.addBodyPart(wrap);

// Define the attachment
MimeBodyPart att = new MimeBodyPart();
DataSource fds = new FileDataSource(ATTACHMENT);
att.setDataHandler(new DataHandler(fds));
att.setFileName(fds.getName());

// Add the attachment to the message.
msg.addBodyPart(att);

// Try to send the email.
try {
 System.out.println("Attempting to send an email through Amazon SES "
 + "using the AWS SDK for Java...");

 // Instantiate an Amazon SES client, which will make the service
 // call with the supplied AWS credentials.
 AmazonSimpleEmailService client =
 AmazonSimpleEmailServiceClientBuilder.standard()
 .withRegion(Regions.US_WEST_2).build();

 // Print the raw email content on the console
 PrintStream out = System.out;
 message.writeTo(out);

 // Send the email.
 ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
 message.writeTo(outputStream);
 RawMessage rawMessage =
 new RawMessage(ByteBuffer.wrap(outputStream.toByteArray()));
 SendRawEmailRequest rawEmailRequest =
new SendRawEmailRequest(rawMessage)
  .withConfigurationSetName(CONFIGURATION_SET);

  client.sendRawEmail(rawEmailRequest);
  System.out.println("Email sent!");
  // Display an error if something goes wrong.
} catch (Exception ex) {
  System.out.println("Email Failed");
  System.err.println("Error message: " + ex.getMessage());
  ex.printStackTrace();
}

Python

The following code example shows how to use the Python email.mime packages and the AWS SDK for Python (Boto) to compose and send a raw email.

```python
import os
import boto3
from botocore.exceptions import ClientError
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.application import MIMEApplication

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
SENDER = "Sender Name <sender@example.com>"

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
RECIPIENT = "recipient@example.com"

# Specify a configuration set. If you do not want to use a configuration
# set, comment the following variable, and the
# ConfigurationSetName=CONFIGURATION_SET argument below.
CONFIGURATION_SET = "ConfigSet"

# If necessary, replace us-west-2 with the AWS Region you're using for Amazon SES.
AWS_REGION = "us-west-2"

# The subject line for the email.
SUBJECT = "Customer service contact info"

# The full path to the file that will be attached to the email.
ATTACHMENT = "path/to/customers-to-contact.xlsx"

# The email body for recipients with non-HTML email clients.
BODY_TEXT = "Hello,
Please see the attached file for a list of customers to contact."

# The HTML body of the email.
BODY_HTML = """"
  <html>
    <head></head>
    <body>
      <h1>Hello!</h1>
      <p>Please see the attached file for a list of customers to contact.</p>
    </body>
  </html>
"""

# The character encoding for the email.
CHARSET = "utf-8"
```
# Create a new SES resource and specify a region.
client = boto3.client('ses',region_name=AWS_REGION)

# Create a multipart/mixed parent container.
msg = MIME_Multipart('mixed')
# Add subject, from and to lines.
msg['Subject'] = SUBJECT
msg['From'] = SENDER
msg['To'] = RECIPIENT

# Create a multipart/alternative child container.
msg_body = MIME_Multipart('alternative')

# Encode the text and HTML content and set the character encoding. This step is
# necessary if you're sending a message with characters outside the ASCII range.
textpart = MIME_Text(BODY_TEXT.encode(CHARSET), 'plain', CHARSET)
htmlpart = MIME_Text(BODY_HTML.encode(CHARSET), 'html', CHARSET)

# Add the text and HTML parts to the child container.
msg_body.attach(textpart)
msg_body.attach(htmlpart)

# Define the attachment part and encode it using MIMEApplication.
att = MIME_Application(open(ATTACHMENT, 'rb').read())

# Add a header to tell the email client to treat this part as an attachment,
# and to give the attachment a name.
att.add_header('Content-Disposition','attachment',filename=os.path.basename(ATTACHMENT))

# Attach the multipart/alternative child container to the multipart/mixed
# parent container.
msg.attach(msg_body)

# Add the attachment to the parent container.
msg.attach(att)

# Provide the contents of the email.
response = client.send_raw_email(
    Source=SENDER,
    Destinations=[
        RECIPIENT
    ],
    RawMessage={
        'Data':msg.as_string(),
    },
    ConfigurationSetName=CONFIGURATION_SET
)

# Display an error if something goes wrong.
except ClientError as e:
    print(e.response['Error']['Message'])
else:
    print("Email sent! Message ID:")
    print(response['ResponseMetadata']['RequestId'])
Sending Personalized Email Using the Amazon SES API

You can use the CreateTemplate API operation to create email templates. These templates include a subject line, and the text and HTML parts of the email body. The subject and body sections may also contain unique values that are personalized for each recipient.

There are a few limits and other considerations when using these features:

- You can create up to 10,000 email templates per Amazon SES account.
- Each template can be up to 500KB in size, including both the text and HTML parts.
- You can include an unlimited number of replacement variables in each template.
- You can send email to up to 50 destinations in each call to the SendBulkTemplatedEmail operation. A destination includes a list of recipients, as well as CC and BCC recipients. Note that the number of destinations you can contact in a single call to the API may be limited by your account’s maximum sending rate. For more information, see Managing Your Amazon SES Sending Limits (p. 127).

This section includes procedures for creating email templates and for sending personalized emails.

Part 1: Set up Rendering Failure Event Notifications (Optional)

If you send an email that contains invalid personalization content, Amazon SES may initially accept the message, but will not be able to deliver it. For this reason, if you plan to send personalized email, you should configure Amazon SES to send Rendering Failure event notifications through Amazon SNS. When you receive a Rendering Failure event notification, you can identify which message contained the invalid content, fix the issues, and send the message again.

The procedure in this section is optional, but highly recommended.

To configure Rendering Failure event notifications

1. Create an Amazon SNS topic. For procedures, see Create a Topic in the Amazon Simple Notification Service Developer Guide.
2. Subscribe to the Amazon SNS topic. For example, if you want to receive Rendering Failure notifications by email, subscribe an email endpoint (that is, your email address) to the topic.
   For procedures, see Subscribe to a Topic in the Amazon Simple Notification Service Developer Guide.
3. Complete the procedures in the section called “Set Up an Amazon SNS Destination” (p. 254) to set up your configuration sets to publish Rendering Failure events to your Amazon SNS topic.

Part 2: Create an Email Template

The instructions in this section assume that you are using the AWS CLI, and that you configured it to interact with your AWS account. For more information about installing and configuring the AWS CLI, see the AWS Command Line Interface User Guide.

To create the template

1. In a text editor, create a new file. Paste the following code into the file.

```json
{
  "Template": {
    "TemplateName": "MyTemplate",
    "SubjectPart": "Greetings, {{name}}!",
```
Part 3: Sending the Personalized Email

After you create an email template, you can use it to send email. There are two API operations that you can use to send emails using templates: SendTemplatedEmail, and SendBulkTemplatedEmail. The SendTemplatedEmail operation is useful for sending a customized email to a single destination (a collection of "To," "CC," and "BCC" recipients who will receive the same email). The SendBulkTemplatedEmail operation is useful for sending unique emails to multiple destinations in a single call to the Amazon SES API. This section provides examples of how to use the AWS CLI to send email using both of these operations.

Sending Templated Email to a Single Destination

You can use the SendTemplatedEmail operation to send an email to a single destination. All of the recipients in the Destination object will receive the same email.

To send a templated email to a single destination

1. In a text editor, create a new file. Paste the following code into the file.

```json
{
   "Source": "sender@example.com",
   "Template": "MyTemplate",
   "ConfigurationSetName": "ConfigSet",
   "Destination": {
      "ToAddresses": [ "alejandro.rosalez@example.com"
   ],
   "TemplateData": "{ "name":"Alejandro", "favoriteanimal": "alligator" }"
}
```

This code contains the following properties:

- **Source** – The email address of the sender.
- **Template** – The name of the template to apply to the email.
- **ConfigurationSetName** – The name of the configuration set to use when sending the email.
Note
We recommend that you use a configuration set that is configured to publish Rendering Failure events to Amazon SNS. For more information, see the section called "Part 1: Set up Notifications (Optional)" (p. 107).

- **Destination** – The recipient addresses. You can include multiple "To," "CC," and "BCC" addresses. When you use the SendTemplatedEmail operation, all recipients receive the same email.

- **TemplateData** – A JSON object that contains key-value pairs. The keys correspond to the variables in the template (for example, \{{\{name\}\}}). The values represent the content that replaces the variables in the email.

2. Change the values in the code above to meet your needs, and then save the file as `myemail.json`.

3. At the command line, type the following command to send the email: `aws ses send-templated-email --cli-input-json file://myemail.json`

**Sending Templated Email to Multiple Destinations**

You can use the SendBulkTemplatedEmail operation to send an email to several destinations in a single call to the API. Amazon SES sends a unique email to the recipient or recipients in each Destination object.

**To send a templated email to multiple destinations**

1. In a text editor, create a new file. Paste the following code into the file.

```json
{
    "Source":"sender@example.com",
    "Template":"MyTemplate",
    "ConfigurationSetName": "ConfigSet",
    "Destinations": [
        {
            "Destination": {
                "ToAddresses": [
                    "anaya.iyengar@example.com"
                ],
                "ReplacementTemplateData": "{ "name":\"Anaya\", "favoriteanimal":\"angelfish\" }"
            },
            {
                "Destination": {
                    "ToAddresses": [
                        "liu.jie@example.com"
                    ],
                    "ReplacementTemplateData": "{ "name":\"Liu\", "favoriteanimal":\"lion\" }"
                },
                {
                    "Destination": {
                        "ToAddresses": [
                            "shirley.rodriguez@example.com"
                        ],
                        "ReplacementTemplateData": "{ "name":\"Shirley\", "favoriteanimal":\"shark\" }"
                    },
                    {
                        "Destination": {
                            "ToAddresses": [
                                "richard.roe@example.com"
                            ],
                            "ReplacementTemplateData": "{ "name":\"Richard\", "favoriteanimal":\"shark\" }"
                        },
                        {
                            "Destination": {
                                "ToAddresses": [
                                    "alan.miller@example.com"
                                ],
                                "ReplacementTemplateData": "{ "name":\"Alan\", "favoriteanimal":\"shark\" }"
                            },
                            {
                                "Destination": {
                                    "ToAddresses": [
                                        "mary.smith@example.com"
                                    ],
                                    "ReplacementTemplateData": "{ "name":\"Mary\", "favoriteanimal":\"shark\" }"
                                },
                                {
                                    "Destination": {
                                        "ToAddresses": [
                                            "jane.doe@example.com"
                                        ],
                                        "ReplacementTemplateData": "{ "name":\"Jane\", "favoriteanimal":\"shark\" }"
                                    },
                                    {
                                        "Destination": {
                                            "ToAddresses": [
                                                "john.doe@example.com"
                                            ],
                                            "ReplacementTemplateData": "{ "name":\"John\", "favoriteanimal":\"shark\" }"
                                        },
                                        {
                                            "Destination": {
                                                "ToAddresses": [
                                                    "bob.smith@example.com"
                                                ],
                                                "ReplacementTemplateData": "{ "name":\"Bob\", "favoriteanimal":\"shark\" }"
                                            },
                                            {
                                                "Destination": {
                                                    "ToAddresses": [
                                                        "sara.perez@example.com"
                                                    ],
                                                    "ReplacementTemplateData": "{ "name":\"Sara\", "favoriteanimal":\"shark\" }"
                                                },
                                                {
                                                    "Destination": {
                                                        "ToAddresses": [
                                                            "mike.wilson@example.com"
                                                        ],
                                                        "ReplacementTemplateData": "{ "name":\"Mike\", "favoriteanimal":\"shark\" }"
                                                    },
                                                    {
                                                        "Destination": {
                                                            "ToAddresses": [
                                                                "jim.green@example.com"
                                                            ],
                                                            "ReplacementTemplateData": "{ "name":\"Jim\", "favoriteanimal":\"shark\" }"
                                                        },
                                                        {
                                                            "Destination": {
                                                                "ToAddresses": [
                                                                    "carlos.montes@example.com"
                                                                ],
                                                                "ReplacementTemplateData": "{ "name":\"Carlos\", "favoriteanimal":\"shark\" }"
                                                            },
                                                            {
                                                                "Destination": {
                                                                    "ToAddresses": [
                                                                        "anna.castillo@example.com"
                                                                    ],
                                                                    "ReplacementTemplateData": "{ "name":\"Anna\", "favoriteanimal":\"shark\" }"
                                                                },
                                                                {
                                                                    "Destination": {
                                                                        "ToAddresses": [
                                                                            "teo.rodriguez@example.com"
                                                                        ],
                                                                        "ReplacementTemplateData": "{ "name":\"Teo\", "favoriteanimal":\"shark\" }"
                                                                    }
                                                                }
                                                            }
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    ]
}
```
This code contains the following properties:

- **Source** – The email address of the sender.
- **Template** – The name of the template to apply to the email.
- **ConfigurationSetName** – The name of the configuration set to use when sending the email.

**Note**

We recommend that you use a configuration set that is configured to publish Rendering Failure events to Amazon SNS. For more information, see the section called "Part 1: Set up Notifications (Optional)" (p. 107).

- **Destinations** – An array that contains one or more Destinations.
  - **Destination** – The recipient addresses. You can include multiple "To," "CC," and "BCC" addresses. When you use the SendBulkTemplatedEmail operation, all recipients within the same Destination object receive the same email.
  - **ReplacementTemplateData** – A JSON object that contains key-value pairs. The keys correspond to the variables in the template (for example, {{name}}). The values represent the content that replaces the variables in the email.
  - **DefaultTemplateData** – A JSON object that contains key-value pairs. The keys correspond to the variables in the template (for example, {{name}}). The values represent the content that replaces the variables in the email. This object contains fallback data. If a Destination object contains an empty JSON object in the ReplacementTemplateData property, the values in the DefaultTemplateData property are used.

2. Change the values in the code above to meet your needs, and then save the file as mybulkemail.json.

3. At the command line, type the following command to send the bulk email: **aws ses send-bulk-templated-email --cli-input-json file://mybulkemail.json**

### Advanced Email Personalization

The template feature in Amazon SES is based on the Handlebars template system. You can use Handlebars to create templates that include advanced features, such as nested attributes, array iteration, basic conditional statements, and the creation of inline partials. This section provides examples of these features.

Handlebars includes additional features beyond those documented in this section. For more information, see Built-In Helpers at handlebarsjs.com.

**Topics**

- Parsing Nested Attributes (p. 110)
- Iterating Through Lists (p. 111)
- Using Basic Conditional Statements (p. 113)
- Creating Inline Partial (p. 114)

### Parsing Nested Attributes

Handlebars includes support for nested paths, which makes it easy to organize complex customer data, and then refer to that data in your email templates.
For example, you can organize recipient data into several general categories. Within each of those categories, you can include detailed information. The following code example shows an example of this structure for a single recipient:

```
{
  "meta": {
    "userId": "51806220607"
  },
  "contact": {
    "firstName": "Anaya",
    "lastName": "Iyengar",
    "city": "Bengaluru",
    "country": "India",
    "postalCode": "560052"
  },
  "subscription": [
    {
      "interest": "Sports"
    },
    {
      "interest": "Travel"
    },
    {
      "interest": "Cooking"
    }
  ]
}
```

In your email templates, you can refer to nested attributes by providing the name of the parent attribute, followed by a period (.), followed by the name of the attribute for which you want to include the value. For example, if you use the data structure shown in the preceding example, and you want to include each recipient's first name in the email template, include the following text in your email template: Hello {{Contact.FirstName}}!

Handlebars can parse paths that are nested several levels deep, which means you have flexibility in how you structure your template data.

**Iterating Through Lists**

The `each` helper function iterates through items in an array. The following code is an example of an email template that uses the `each` helper function to create an itemized list of each recipient's interests.

```
{
  "TemplateName": "Preferences",
  "SubjectPart": "Subscription Preferences for {{contact.firstName}}{{contact.lastName}}",
  "HtmlPart": "<h1>Your Preferences</h1><p>You have indicated that you are interested in receiving information about the following subjects:</p><ul>
{{#each subscription}}
  <li>{{interest}}</li>
{{/each}}
</ul><p>You can change these settings at any time by visiting the <a href=https://www.example.com/preferences/i.aspx?id={{meta.userId}}>Preference Center</a>.</p>
"TextPart": "Your Preferences

You have indicated that you are interested in receiving information about the following subjects:

{{#each subscription}}
"
Important

In the preceding code example, the values of the `HtmlPart` and `TextPart` attributes include line breaks to make the example easier to read. The JSON file for your template can't contain line breaks within these values. If you copied and pasted this example into your own JSON file, remove the line breaks and extra spaces from the `HtmlPart` and `TextPart` sections before proceeding.

After you create the template, you can use the `SendTemplatedEmail` or the `SendBulkTemplatedEmail` operation to send email to recipients using this template. As long as each recipient has at least one value in the `Interests` object, they receive an email that includes an itemized list of their interests. The following example shows a JSON file that can be used to send email to multiple recipients using the preceding template:

```json
{
    "Source": "Sender Name <sender@example.com>",
    "Template": "Preferences",
    "Destinations": [
        {
            "Destination": {
                "ToAddresses": [
                    "anaya.iyengar@example.com"
                ],
                "ReplacementTemplateData": "{"meta": {"userId": "51806220607"}, "contact": {
                    "firstName": "Anaya",
                    "lastName": "Iyengar"
                }, "subscription": [{"interest": "Sports"}, {"interest": "Travel"}, {"interest": "Cooking"}]}}",
            "Destination": {
                "ToAddresses": [
                    "shirley.rodriguez@example.com"
                ],
                "ReplacementTemplateData": "{"meta": {"userId": "1981624758263"}, "contact": {
                    "firstName": "Shirley",
                    "lastName": "Rodriguez"
                }, "subscription": [{"interest": "Technology"}, {"interest": "Politics"}]}}",
            "DefaultTemplateData": "{"meta": "", "contact": {"firstName": "Friend", "lastName": ""}, "subscription": []}"
        }
    ],
    "DefaultTemplateData": "{"meta": "", "contact": {"firstName": "Friend", "lastName": ""}, "subscription": []}
}
```

When you send an email to the recipients listed in the preceding example using the `SendBulkTemplatedEmail` operation, they receive a message that resembles the example shown in the following image:
Using Basic Conditional Statements

This section builds on the example described in the previous section. The example in the previous section uses the each helper to iterate through a list of interests. However, recipients for whom no interests are specified receive an email that contains an empty list. By using the {{if}} helper, you can format the email differently if a certain attribute is present in the template data. The following code uses the {{if}} helper to display the bulleted list from the preceding section if the Subscription array contains any values. If the array is empty, a different block of text is displayed.

```json
{
    "TemplateName": "Preferences2",
    "SubjectPart": "Subscription Preferences for {{contact.firstName}} {{contact.lastName}}",
    "HtmlPart": "<h1>Your Preferences</h1>
               <p>Dear {{contact.firstName}},</p>
               {{#if subscription}}
                   <p>You have indicated that you are interested in receiving information about the following subjects:</p>
                   <ul>
                   {{#each subscription}}
                   <li>{{interest}}</li>
                   {{/each}}
                   </ul>
                   <p>You can change these settings at any time by visiting the <a href=https://www.example.com/preferences/i.aspx?id={{meta.userId}}>Preference Center</a>.</p>
               {{else}}
                   <p>Please update your subscription preferences by visiting the <a href=https://www.example.com/preferences/i.aspx?id={{meta.userId}}>Preference Center</a>.</p>
               {{/if}}",
    "TextPart": "Your Preferences

Dear {{contact.firstName}},

{{#if subscription}}
You have indicated that you are interested in receiving information about the following subjects:

- {{interest}}

You can change these settings at any time by visiting the Preference Center at https://www.example.com/preferences/i.aspx?id={{meta.userId}}.
{{else}}
Please update your subscription preferences by visiting the Preference Center at https://www.example.com/preferences/i.aspx?id={{meta.userId}}.
{{/if}}"
Important

In the preceding code example, the values of the HtmlPart and TextPart attributes include line breaks to make the example easier to read. The JSON file for your template can’t contain line breaks within these values. If you copied and pasted this example into your own JSON file, remove the line breaks and extra spaces from the HtmlPart and TextPart sections before proceeding.

The following example shows a JSON file that can be used to send email to multiple recipients using the preceding template:

```
{
    "Source":"Sender Name <sender@example.com>",
    "Template":"Preferences2",
    "Destinations": [
        {
            "Destination": {
                "ToAddresses": [
                    "anaya.iyengar@example.com"
                ],
                "ReplacementTemplateData": "{"meta": {"userId": "51806220607"}, "contact": {
                "firstName": "Anaya",
                "lastName": "Iyengar",
                "subscription": [{"interest": "Sports"}, {"interest": "Cooking"}]}}"
            },
            "Destination": {
                "ToAddresses": [
                    "shirley.rodriguez@example.com"
                ],
                "ReplacementTemplateData": "{"meta": {"userId": "1981624758263"}, "contact": {
                "firstName": "Shirley",
                "lastName": "Rodriguez"
            }}"
            }
        ],
        "DefaultTemplateData": "{"meta": {"userId": ""}, "contact": {"firstName": "Friend", "lastName": ""}, "subscription": []}
    }
}
```

In this example, the recipient whose template data included a list of interests receives the same email as the example shown in the previous section. The recipient whose template data did not include any interests, however, receives an email that resembles the example shown in the following image:

![Your Preferences](image)

Creating Inline Partials

You can use inline partials to simplify templates that include repeated strings. For example, you could create an inline partial that includes the recipient's first name, and, if it is available, their last name by adding the following code to the beginning of your template:

```
{{#* inline "fullName"}}{{firstName}}{{#if lastName}} {{lastName}}{{/if}}{{/inline}}
```

Note

The newline character (\n) is required to separate the {{inline}} block from the content in your template. The newline isn't rendered in the final output.
After you create the `fullName` partial, you can include it anywhere in your template by preceding the name of the partial with a greater-than (>) sign followed by a space, as in the following example: `{{> fullName}}`. Inline partials are not transferred between parts of the email. For example, if you want to use the same inline partial in both the HTML and the text version of the email, you must define it in both the `HtmlPart` and the `TextPart` sections.

You can also use inline partials when iterating through arrays. You can use the following code to create a template that uses the `fullName` inline partial. In this example, the inline partial applies to both the recipient's name and to an array of other names:

```json
{
    "TemplateName": "Preferences3",
    "SubjectPart": "{{firstName}}'s Subscription Preferences",
    "HtmlPart": "{{#* inline "fullName"}}
        {{firstName}}{{#if lastName}} {{lastName}}{{/if}}
    {{/inline~}}

    <h1>Hello {{> fullName}}!</h1>
    <p>You have listed the following people as your friends:</p>
    <ul>
        {{#each friends}}
        <li>{{> fullName}}</li>
    {{/each}}</ul>
}
```

Important
In the preceding code example, the values of the `HtmlPart` and `TextPart` attributes include line breaks to make the example easier to read. The JSON file for your template can't contain line breaks within these values. If you copied and pasted this example into your own JSON file, remove the line breaks and extra spaces from these sections.

Authenticating Your Email in Amazon SES

Amazon Simple Email Service (Amazon SES) uses the Simple Mail Transfer Protocol (SMTP) to send email. Because SMTP does not provide any authentication by itself, spammers can send email messages that claim to originate from someone else, while hiding their true origin. By falsifying email headers and spoofing source IP addresses, spammers can mislead recipients into believing that the email messages that they are receiving are authentic.

Most ISPs that forward email traffic take measures to evaluate whether email is legitimate. One such measure that ISPs take is to determine whether an email is authenticated. Authentication requires senders to verify that they are the owner of the account that they are sending from. In some cases, ISPs refuse to forward email that is not authenticated. To ensure optimal deliverability, we recommend that you authenticate your emails.

The following sections describe two authentication mechanisms ISPs use—Sender Policy Framework (SPF) and DomainKeys Identified Mail (DKIM)—and provide instructions for how to use these standards with Amazon SES.
• To learn about SPF, which provides a way to trace an email message back to the system from which it was sent, see Authenticating Email with SPF in Amazon SES (p. 116).

• To learn about DKIM, a standard that allows you to sign your email messages to show ISPs that your messages are legitimate and have not been modified in transit, see Authenticating Email with DKIM in Amazon SES (p. 117).

• To learn how to comply with Domain-based Message Authentication, Reporting and Conformance (DMARC), which relies on SPF and DKIM, see Complying with DMARC Using Amazon SES (p. 126).

Authenticating Email with SPF in Amazon SES

Sender Policy Framework (SPF) is an email validation standard, defined in RFC 7208, designed to combat email spoofing. SPF enables domain owners to specify which mail servers are authorized to send email for their domain. To indicate compliance with SPF, the domain owner publishes a list of authorized mail servers in a DNS record on the domain’s DNS server. When a receiving mail server receives an email that contains the domain in the MAIL FROM address, it checks the domain’s DNS records to compare the sending mail server to the authorized mail servers and takes action on the email accordingly.

An SPF record indicates to ISPs that you have authorized Amazon SES to send email for your domain. When you use Amazon SES, your decision about whether to publish an SPF record depends on whether you only require your email to pass an SPF check by the receiving mail server, or if you want your email to comply with the additional requirements needed to pass Domain-based Message Authentication, Reporting and Conformance (DMARC) authentication based on SPF. You can use DKIM to achieve DMARC validation, but it is a best practice to use both DKIM and SPF for maximum deliverability.

• To pass an SPF check—When you use Amazon SES, there are two setups with which you can pass an SPF check. The first setup is to use the default MAIL FROM domain of Amazon SES, and to not publish an SPF record at all. This setup enables you to pass an SPF check because by default, Amazon SES uses its own MAIL FROM domain to send your emails. In this case, an SPF check will pass because the default MAIL FROM domain is amazonses.com (or a subdomain of that) and the sending mail server is Amazon SES.

The other setup with which you can pass an SPF check is to configure Amazon SES to use your own MAIL FROM domain, in which case you must publish an SPF record because the MAIL FROM domain and the domain of the sending mail server, Amazon SES, are different. Instructions for configuring your domain to send emails using a custom MAIL FROM domain are provided in Using a Custom MAIL FROM Domain (p. 57).

• To pass DMARC validation based on SPF—If you want DMARC validation to succeed based on SPF, you must set up a custom MAIL FROM domain (p. 57) and publish an SPF record. Note that the alignment mode in the DMARC policy must be relaxed, which is the default. For more information about DMARC policies, see https://dmarc.org/.

Adding an SPF Record

The procedure for adding a TXT record to your domain’s DNS settings depends on who provides your DNS service, but for general instructions, see Adding a TXT Record to Your Domain’s DNS Server in Amazon SES Domain Verification TXT Records (p. 56). For information specific to SPF records, go to http://www.openspf.net and RFC 7208.

Adding a New SPF Record

If your custom MAIL FROM domain does not have an existing SPF record, publish a TXT record with the following value. The name of the record can be blank or @, depending on your DNS service.

Important

If you use "-all" as shown in the example, ISPs might block email from IP addresses that are not listed in your SPF record. Your SPF record must therefore include every IP address that you
use to send email. As a debugging aid, you can use "~all" instead. When you use "~all", ISPs will typically accept email from IP addresses that are not listed in the SPF record, but they might flag it. To maximize deliverability, use "~all" and add a record for each IP address. For examples of how to authorize multiple IP addresses, go to http://www.openspf.org/SPF_Record_Syntax.

```
"v=spf1 include:amazonses.com -all"
```

Adding to an Existing SPF Record

If your domain already has an SPF record, then you must add the following SPF mechanism to the existing record.

```
include:amazonses.com
```

Authenticating Email with DKIM in Amazon SES

DomainKeys Identified Mail (DKIM) is a standard that allows senders to sign their email messages and ISPs to use those signatures to verify that those messages are legitimate and have not been modified by a third party in transit.

An email message that is sent using DKIM includes a DKIM-Signature header field that contains a cryptographically signed representation of all, or part, of the message. An ISP receiving the message can decode the cryptographic signature using a public key, published in the sender's DNS record, to ensure that the message is authentic. For more information about DKIM, see http://www.dkim.org.

DKIM signatures are optional. You might decide to sign your email using a DKIM signature to enhance deliverability with DKIM-compliant ISPs. Amazon SES provides two options to sign your messages using a DKIM signature:

- To set up your domain so that Amazon SES automatically adds a DKIM signature to every message sent from that domain, see Easy DKIM in Amazon SES (p. 117).
- To add your own DKIM signature to any email that you send using the SendRawEmail API, see Manual DKIM Signing in Amazon SES (p. 125).

Easy DKIM in Amazon SES

Easy DKIM is a feature of Amazon SES that signs every message that you send from a verified email address or domain with a DKIM signature that uses a 1024-bit DKIM key. You can use the Amazon SES console to configure Easy DKIM settings, and to enable or disable automatic DKIM signing for your email messages. You must be able to edit your domain's DNS records to set up Easy DKIM. With the appropriate DNS records in place, you can enable Easy DKIM signing for any verified email address or domain.

The following rules apply:

- As with other verified identity settings, if you verify a domain, subdomain, and email address that share a root domain, Easy DKIM settings apply at the most granular level you verified. That is:
  - Verified email address settings override verified domain settings.
  - Verified subdomain settings override verified domain settings, with lower-level subdomain settings overriding higher-level subdomain settings.
  - If you verify a root domain, and do not verify a particular subdomain, then that subdomain uses the root domain's Easy DKIM settings. That is, if the root domain has Easy DKIM set up and enabled, the subdomain's emails are automatically DKIM-signed as well.
Once you set up Easy DKIM, your messages will automatically be DKIM-signed regardless of whether you call Amazon SES through the SMTP interface or the API (SendEmail or SendRawEmail). Note that you only need to set up Easy DKIM for the domain you use in your From address, not for the domain in a “Return-Path” or “Reply-To” address.

If you are verifying a new domain, you can set up Easy DKIM at that time. If you already have a verified domain or email address, you can add Easy DKIM capability to it whenever you want.

**Note**
Amazon SES has endpoints in multiple AWS regions, and Easy DKIM setup applies to each AWS region separately. You must perform the Easy DKIM setup procedure for each region in which you want to use Easy DKIM. For information about using Amazon SES in multiple AWS regions, see Regions and Amazon SES (p. 409).

This topic contains the following sections:

- To set up Easy DKIM while you verify a new domain, see Setting Up Easy DKIM for a New Domain (p. 118).
- To set up Easy DKIM for an email address or domain that you have already verified, see Setting Up Easy DKIM for an Existing Verified Identity (p. 120).

### Setting Up Easy DKIM for a New Domain

When you use the AWS Management Console to verify a new domain, you can also set up Easy DKIM at the same time.

These instructions are for new domains only. If you want to set up Easy DKIM for an email address or domain that you have already verified, see Setting Up Easy DKIM for an Existing Verified Identity (p. 120).

**To set up Easy DKIM for a new domain**

1. Go to your verified domain list in the Amazon SES console, or follow these instructions to navigate to it:
   a. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
   b. In the navigation pane, under Identities, click Domains.
2. Click Verify a New Domain.
3. In the Verify a New Domain dialog box, enter your domain name, select the Generate DKIM settings check box, and then click Verify This Domain.
In the resulting dialog box, you will see all of the DNS records that you need for setting up domain verification and Easy DKIM. This information will also be available by clicking the domain name after you close the dialog box.

4. To complete domain verification, you must update your domain's DNS settings with the TXT record information from the **Domain Verification Record** in the **Verify a New Domain** dialog box. Note that some domain name providers use the term **Host** instead of **Name**. If your DNS provider does not allow underscores in record names, you can omit _amazonses from the **Name** of the domain verification record. To help you easily identify this record within your domain's DNS settings, you can optionally prefix the **Value** with amazonses:

Highlight and copy individual records, or select **Download Record Set as CSV** to download all of the records.

**Important**

DNS providers may append the domain name to the end of DNS records. Adding a record that already contains the domain name (such as _amazonses.example.com) may result in the duplication of the domain name (such as _amazonses.example.com.example.com). To avoid duplication of the domain name, add a period to the end of the domain name in the DNS record. This will indicate to your DNS provider that the record name is fully qualified (that is, no longer relative to the domain name), and prevent the DNS provider from appending an additional domain name.

5. To set up DKIM, you must update your domain's DNS settings with the CNAME record information from the dialog box. Note that you cannot omit the underscore from _domainkey because the underscore is required by RFC 4871.

Highlight and copy individual CNAME records, or select **Download Record Set as CSV** to download all of the records.

a. If Route 53 provides the DNS service for the domain you are verifying, and you are logged in to Amazon SES console with the same email address and password you use for Route 53, then you will have the option of immediately updating your DNS settings for both domain verification and DKIM from within the Amazon SES console.

b. If you are not using Route 53, you will need to update your DNS settings according to the procedure established by your DNS service provider. (Ask your system administrator if you are
not sure who provides your DNS service.) Amazon Web Services will eventually detect that you have updated your DNS records; this detection process may take up to 72 hours.

When verification is complete, the domain's **Status** in the Amazon SES console will change from *pending verification* to *verified*, and you will receive an *Amazon SES Domain Verification SUCCESS* confirmation email from Amazon Web Services. (AWS emails are sent to the email address you used when you signed up for Amazon SES.)

When Amazon SES has successfully detected the changes to your DNS records, the **DKIM Verification Status** for that domain in the Amazon SES console will change from *in progress* to *success*, and you will receive an *Amazon SES DKIM Setup Successful* confirmation email from Amazon Web Services.

6. You can now use Amazon SES to send email that is signed using a DKIM signature from any valid address in the verified domain. To send a test email using the Amazon SES console, check the box next to the verified domain, and then click **Send a Test Email**. View the email headers in the email you receive. Email providers typically provide this capability through an option such as *Show original* or *View message source*. Look for a header named **DKIM-Signature** with the "d" tag set to your domain. Note that when DKIM is enabled, there will be two **DKIM-Signature** headers added to the message: one header for your domain, and one header with d=amazonses.com. (Amazon SES adds a signature for amazonses.com automatically whether you have DKIM enabled or not. You can ignore it.) For example, for a domain called *ses-example.com*, the DKIM signature header you are looking for might look like:

```
DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/simple;
    s=ses-example.com; d=ses-example.com;
    t=1366720445;
    h=From:To:Subject:MIME-Version:Content-Type:Content-Transfer-Encoding:Date:Message-ID;
    bh=lcj/Sl5qKl6K6zwFWUwb7Flgnngl892pW574kmS1hrS0=;
    b=nhVMQlmSh7/DM5FW7xPV4k/PN4iYVoa5OF4YYkz2L7jgQ9hHQlckpoxe82TaAr64
    eVTcBhHfj9Bwtxkmuk88g6UUN8J+AAad/JUNGoZOBStloofSkuA96cGfRGanF68Ag7
    nmm8jKi+JL5QhQ/+/EKTH4TVb4zdEW1BuMr4dTg=*
```

**Important**

How you update the DNS settings depends on who provides your DNS service. DNS service may be provided by a domain name registrar such as GoDaddy or Network Solutions, or by a separate service such as Amazon Route 53.

**What if Easy DKIM fails?**

If your DNS settings are not correctly updated, you will first receive an *Amazon SES DKIM FAILURE* email from Amazon Web Services, and you will see a status of *failed* in the Domains area when you click on the DKIM tab.

**Note**

If this happens, Amazon SES will still send your email, but it will *not be signed using a DKIM signature*.

**Setting Up Easy DKIM for an Existing Verified Identity**

If you have already verified a domain or email address, you can use the AWS Management Console to set up Easy DKIM for that identity at any time.

These instructions are for adding DKIM signing to a domain that has already been verified. If you are verifying a new domain and want to set up Easy DKIM at the same time, see Setting Up Easy DKIM for a New Domain (p. 118).
Important
Easy DKIM only works with fully qualified domain names (FQDNs). If you wanted to set up Easy DKIM for both example.com and newyork.example.com, you would need to set up Easy DKIM for each of these FQDNs separately.

To set up Easy DKIM for an existing verified domain

1. Go to your verified domain list in the Amazon SES console, or follow these instructions to navigate to it:
   a. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
   b. In the navigation pane, under Identities, click Domains.
2. In the content pane, click the verified domain for which you would like to set up Easy DKIM.
3. On the domain's Details page, expand DKIM.
4. Click Generate DKIM Settings.

   Your DKIM records will be displayed.
5. To set up DKIM, you must update your domain's DNS settings with the displayed CNAME record information. You can copy the records or click the Download Record Set as CSV link.
   a. If Route 53 provides the DNS service for the domain you are verifying, and you are logged in to Amazon SES console with the same email address and password you use for Route 53, then Amazon SES will give you the option of immediately updating your DNS settings for Easy DKIM. If you would like to do this, click the Use Route 53 button.

   Next, click Create Record Sets in the Use Route 53 dialog box to complete the process.

   b. If you are not using Route 53, you will need to update your DNS settings according to the procedure established by your DNS service provider. (Ask your system administrator if you are not sure who provides your DNS service.) Amazon Web Services will eventually detect that you have updated your DNS records; this detection process may take up to 72 hours.

6. When Amazon SES has successfully detected the changes to your DNS records, the DKIM Verification Status for that domain in the Amazon SES console will change from in progress to success, and you will receive an Amazon SES DKIM Setup Successful confirmation email from Amazon Web Services. (Amazon Web Services emails are sent to the email address you used when you signed up for Amazon SES.)

7. (This step is only required if DKIM setup was initiated before 09-13-16, 2:00 PDT) To sign your messages using a DKIM signature, you must enable Easy DKIM for the appropriate verified sending identity as follows:
   a. In the navigation pane, under Identities, click either Email Addresses or Domains, depending whether you want to enable Easy DKIM signing for an email address or a domain.
   b. Click the email address or domain for which you wish to enable Easy DKIM signing.
   c. On the Details page of the email address or domain, expand DKIM.
   d. In the DKIM: field, click enable.

8. You can now use Amazon SES to send email that is signed using a DKIM signature from any valid address in the verified domain. To send a test email using the Amazon SES console, check the box next to the verified domain, and then click Send a Test Email. View the email headers in the email you receive. Email providers typically provide this capability through an option such as Show original or View message source. Look for a header named DKIM-Signature with the "d" tag set to your domain. Note that when DKIM is enabled, there will be two DKIM-Signature headers added to the message: one header for your domain, and one header with d=amazonses.com. (Amazon SES adds a signature for amazonses.com automatically whether you have DKIM enabled or not. You can ignore it.) For example, for a domain called ses-example.com, the DKIM signature header you are looking for might look like:
DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/simple; s=xtk53kxcy4p3t6zbrfss6d54rerrrh6; d=ses-example.com; t=1366720445; h=From:To:Subject:MIME-Version:Content-Type:Content-Transfer-Encoding:Date:Message-ID; bh=lcj/Sl5qE16K6swF1wb7Plgng1l892pW574kmS1hrS0=; b=nhVMQLmSh7/DM5FW7xPV4k/PN41VYo0a50F4YyK2L7jqUg9hQ1ckopxe82TaAr64eVTCbHtHj9Bwtkmuk8g4G5UUNj+AAad/JUNGozOBSiOofSkuAQ6cGfRGanF68Ag7nnmEjXi+JL5JQh/ueXTH4TVb4zdEw1BuMlrdTg=

Important
How you update the DNS settings depends on who provides your DNS service. DNS service may be provided by a domain name registrar such as GoDaddy or Network Solutions, or by a separate service such as Route 53.

What if Easy DKIM fails?
If your DNS settings are not correctly updated, you will first receive an Amazon SES DKIM FAILURE email from Amazon Web Services, and you will see a status of failed in the Domains area when you click on the DKIM tab.

Note
If this happens, Amazon SES will still send your email, but it will not be signed using a DKIM signature.

Disabling Easy DKIM in Amazon SES
If you want to temporarily stop Amazon SES from signing your messages using DKIM, you can disable Easy DKIM for your email address or domain. You can reenable it at any time.

To disable Easy DKIM signing
1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation pane, under Identities, click either Email Addresses or Domains, depending whether you want to disable Easy DKIM signing for an email address or a domain.
3. Click the email address or domain for which you wish to disable Easy DKIM signing.
4. On the Details page of the email address or domain, expand DKIM.
5. In the DKIM: field, click disable. Amazon SES will no longer DKIM-sign emails that you send from this identity.

Note
If you do not see the disable option as in the figure below, then DKIM is already disabled.
Note
If you want to permanently disable DKIM signing from any email address on that domain, you should also remove the CNAME records from your DNS.

DKIM Record Revocation in Amazon SES

Amazon Web Services periodically reviews DKIM DNS records, and revokes DKIM signing in cases where it is no longer valid. If Amazon Web Services is unable to detect the CNAME record information required to confirm the ownership of a domain, you will receive an Amazon SES DKIM REVOCATION WARNING email from Amazon Web Services. Amazon SES will continue to send your email, but it will not be signed using a DKIM signature.

If you restore the CNAME record information to your DNS settings within five days, you will receive an Amazon SES DKIM REVOCATION CANCELLATION email from Amazon Web Services. Amazon SES will once again sign email using a DKIM signature from a verified identity for which you have enabled Easy DKIM.

If you do not restore the CNAME record information to your DNS settings within five days, you will receive an Amazon SES DKIM REVOCATION email from Amazon Web Services, and email you send via Amazon SES will not be signed using a DKIM signature.

To set up Easy DKIM for a domain for which DKIM signing has been revoked, you must restart the procedure from the beginning, as if you were setting up Easy DKIM for the first time.

Other Ways to Manage Easy DKIM in Amazon SES

You can also manage Easy DKIM with the Amazon SES API. The following actions are available:

- VerifyDomainDkim
- SetIdentityDkimEnabled
- GetIdentityDkimAttributes

You can use these API actions to write a customized front-end application for working with Easy DKIM. For a complete description of API actions related to Easy DKIM, go to the Amazon Simple Email Service API Reference.
Creating DNS Records for DKIM Signing in Amazon SES

Unlike the Amazon SES console, the Amazon SES API does not generate fully-formed DNS records for use with DKIM. Instead, they return DKIM tokens — character strings that represent your domain's identity.

If you are not using the Amazon SES console, you will need to create your own CNAME records using the DKIM tokens returned by the API.

To create DNS records for DKIM signing

1. Obtain the DKIM tokens for your domain. To do so, if you are using the Amazon SES API, call VerifyDomainDkim to generate the tokens. If you already have a DKIM verified identity, call GetIdentityDkimAttributes to obtain the tokens.

2. In the output from the API, you will receive three DKIM tokens similar to the following:

   vvjuipp74whm76gqoni7qmwwn4w4qusjiaiinvf6sf
   3frqe7jn4obpxjwpolz6ipb3k5nvt2nhjpik2oy
   wrqpleh7oodxmad7hs14mxg2avzneazxv5xi2

3. Use these tokens to construct three CNAME records. For a domain named example.com, the records should appear similar to these:

   vvjuipp74whm76gqoni7qmwwn4w4qusjiaiinvf6sf._domainkey.example.com CNAME
   vvjuipp74whm76gqoni7qmwwn4w4qusjiaiinvf6sf.dkim.amazonses.com
   3frqe7jn4obpxjwpolz6ipb3k5nvt2nhjpik2oy._domainkey.example.com CNAME
   3frqe7jn4obpxjwpolz6ipb3k5nvt2nhjpik2oy.dkim.amazonses.com
   wrqpleh7oodxmad7hs14mxg2avzneazxv5xi2._domainkey.example.com CNAME
   wrqpleh7oodxmad7hs14mxg2avzneazxv5xi2.dkim.amazonses.com

You can now update your DNS with these records. Amazon Web Services will eventually detect that you have updated your DNS records; this detection process may take up to 72 hours. Upon successful detection, you will receive an Amazon SES DKIM Setup Successful confirmation email from Amazon Web Services. (Amazon Web Services emails are sent to the email address you used when you signed up for Amazon SES.)

Manual DKIM Signing in Amazon SES

If you prefer not to use Easy DKIM, you can still sign your email messages using a DKIM signature and send them using Amazon SES. To do this, you must use the SendRawEmail API and self-sign your message content according to the specifications provided at http://www.dkim.org. If you use this approach, be aware that Amazon SES does not validate the DKIM signature that you construct. If there are any errors in the signature, you will need to correct them yourself. If you DKIM-sign your own email messages, we recommend that you use keys that are at least 1024 bits.

Whether or not you DKIM-sign your messages, Amazon SES automatically adds a DKIM header with d=amazones.com, which you can ignore. If you do DKIM-sign your messages, it is expected that there will be two DKIM headers: one for your domain, and one for amazones.com.

Important
To ensure maximum deliverability, do not sign any of the following headers using a DKIM signature:

- Message-ID
- Date
- Return-Path
- Bounces-To
Note
If you are using the Amazon SES SMTP interface to send email, and your client software automatically performs DKIM signing, you should check to ensure that your client does not sign any of the headers listed above. We recommend that you check the documentation for your software to find out exactly what headers are signed with DKIM.
For more information about the Amazon SES SMTP interface, see Using the Amazon SES SMTP Interface to Send Email (p. 66).

Complying with DMARC Using Amazon SES

Domain-based Message Authentication, Reporting and Conformance (DMARC) is an email authentication protocol that uses Sender Policy Framework (SPF) and DomainKeys Identified Mail (DKIM) to detect email spoofing. In order to comply with DMARC, messages must be authenticated through either SPF or DKIM, or both.

This topic contains information that will help you configure Amazon SES so that the emails you send comply with both SPF and DKIM. By complying with one of these authentication systems, your emails will comply with DMARC. For information about the DMARC specification, see http://www.dmarc.org.

Complying with DMARC through SPF

For an email to comply with DMARC based on SPF, both of the following conditions must be met:

- The email must pass an SPF check.
- The domain in the From address of the email header must align with the MAIL FROM domain that the sending mail server specifies to the receiving mail server. If the domain's DMARC policy for SPF specifies strict alignment, the From and MAIL FROM domains must match exactly. If the domain's DMARC policy for SPF specifies relaxed alignment, the MAIL FROM domain can be a subdomain of the domain in the From header.

To comply with these requirements, complete the following steps:

- Set up a custom MAIL FROM domain by completing the procedures in the section called "Using a Custom MAIL FROM Domain" (p. 57).
- Ensure that your sending domain uses a relaxed policy for SPF. If you have not changed your domain's policy alignment, it will use a relaxed policy by default.

Note
You can determine your domain's DMARC alignment for SPF by typing the following command at the command line, replacing example.com with your domain:

```
nslookup -type=TXT _dmarc.example.com
```

In the output of this command, under Non-authoritative answer, look for a record that begins with v=DMARC1. If this record includes the string aspf=r, or if the aspf string is not present at all, then your domain uses relaxed alignment for SPF. If the record includes the string aspf=s, then your domain uses strict alignment for SPF. Your system administrator will need to remove this tag from the DMARC TXT record in your domain's DNS configuration. Alternatively, you can use a web-based DMARC lookup tool, such as the DMARC Inspector from dmarcian.com, to determine your domain's policy alignment for SPF.

Complying with DMARC through DKIM

For an email to comply with DMARC based on DKIM, both of the following conditions must be met:
• The message must have a valid DKIM signature.
• The From address in the email header must align with the \texttt{d=} domain in the DKIM signature. If the domain's DMARC policy specifies strict alignment for DKIM, these domains must match exactly. If the domain's DMARC policy specifies relaxed alignment for DKIM, the \texttt{d=} domain can be a subdomain of the From domain.

To comply with these requirements, complete the following steps:

• Set up Easy DKIM by completing the procedures in the section called “Easy DKIM” (p. 117). When you use Easy DKIM, Amazon SES will automatically sign your emails.

  \textbf{Note}
  Rather than use Easy DKIM, you can also manually sign your messages (p. 125). However, you must be very careful if you choose to do so, because Amazon SES does not validate the DKIM signature that you construct. For this reason, we highly recommend using Easy DKIM.

• Ensure that your sending domain uses a relaxed policy for DKIM. If you have not changed your domain's policy alignment, it will use a relaxed policy by default.

  \textbf{Note}
  You can determine your domain's DMARC alignment for DKIM by typing the following command at the command line, replacing \texttt{example.com} with your domain:

  \begin{verbatim}
  nslookup -type=TXT _dmarc.example.com
  \end{verbatim}

  In the output of this command, under \textbf{Non-authoritative answer}, look for a record that begins with \texttt{v=DMARC1}. If this record includes the string \texttt{adkim=r}, or if the \texttt{adkim} string is not present at all, then your domain uses relaxed alignment for DKIM. If the record includes the string \texttt{adkim=a}, then your domain uses strict alignment for DKIM. Your system administrator will need to remove this tag from the DMARC TXT record in your domain's DNS configuration. Alternatively, you can use a web-based DMARC lookup tool, such as the DMARC Inspector from dmarcian.com, to determine your domain's policy alignment for DKIM.

\section*{Managing Your Amazon SES Sending Limits}

Your Amazon Simple Email Service (Amazon SES) account has a set of sending limits to regulate the number of email messages that you can send and the rate at which you can send them. Sending limits benefit all Amazon SES customers because they help to maintain the trusted relationship between Amazon SES and internet service providers (ISPs). Sending limits help you to gradually ramp up your sending activity and decrease the likelihood that ISPs will block your emails because of sudden, unexpected spikes in your email sending volume or rate.

The following are Amazon SES sending limits:

• \textbf{Sending Quota}—The maximum number of emails that you can send in a 24-hour period. The sending quota reflects a rolling time period. Every time you try to send an email, Amazon SES checks how many emails you sent in the previous 24 hours. As long as the total number of emails that you have sent is less than your quota, your send request will be accepted and your email will be sent. If you have already sent your full quota, your send request will be rejected with a throttling exception. For example, if your sending quota is 50,000, and you sent 15,000 emails in the previous 24 hours, then you can send another 35,000 emails right away. If you have already sent 50,000 emails in the previous 24 hours, you will not be able to send more emails until some of the previous sending rolls out of its 24-hour window.

• \textbf{Maximum Send Rate}—The maximum number of emails that Amazon SES can accept from your account per second. You can exceed this limit for short bursts, but not for a sustained period of time.
Note
The rate at which Amazon SES accepts your messages might be less than the maximum send rate.

Your Amazon SES sending limits for each AWS region are separate. For information about using Amazon SES in multiple AWS regions, see Regions and Amazon SES (p. 409).

When you are in the Amazon SES sandbox, your sending quota is 200 messages per 24-hour period and your maximum sending rate is one message per second. To increase your sending limits, you need to submit an SES Sending Limits Increase case. For more information, see Moving Out of the Amazon SES Sandbox (p. 64). After you have moved out of the sandbox and start sending emails, you can increase your sending limits further by submitting another SES Sending Limits Increase case, as discussed in Increasing Your Amazon SES Sending Limits (p. 129).

Note
Sending limits are based on recipients rather than on messages. For example, an email that has 10 recipients counts as 10 against your quota. However, we do not recommend that you send an email to multiple recipients in one call to SendEmail because if the call to Amazon SES fails (for example, the request is improperly formatted), the entire email will be rejected and none of the recipients will get the intended email. We recommend that you call SendEmail once for every recipient.

• To increase your sending limits, see Increasing Your Amazon SES Sending Limits (p. 129).
• For information about the errors your application receives when you reach your sending limits, see What Happens When You Reach Your Amazon SES Sending Limits (p. 131).
• To monitor your sending limits by using the Amazon SES console or the Amazon SES API, see Monitoring Your Amazon SES Sending Limits (p. 128).

Monitoring Your Amazon SES Sending Limits

You can monitor your sending limits by using the Amazon SES console or through the Amazon SES API, whether by calling the Query (HTTPS) interface directly or indirectly through an AWS SDK, the AWS Command Line Interface, or the AWS Tools for Windows PowerShell.

Important
We recommend that you frequently check your sending statistics to ensure that you are not close to your sending limits. If you are close to your sending limits, see Increasing Your Amazon SES Sending Limits (p. 129) for information about how to increase them. Don't wait until you reach your sending limits to consider increasing them.

Monitoring Your Sending Limits Using the Amazon SES Console

The following procedure shows you how to view your sending limits using the Amazon SES console.

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation pane, choose Sending Statistics. Your sending limits are shown under Your Amazon SES Sending Limits.
3. To update the display, choose **Refresh**.

### Monitoring Your Sending Limits Using the Amazon SES API

The Amazon SES API provides the `GetSendQuota` action, which returns your sending limits. When you call `GetSendQuota` action, you receive the following information:

- Number of emails you have sent during the past 24 hours
- Sending quota for the current 24-hour period
- Maximum send rate

**Note**
For a description of `GetSendQuota`, see [Amazon Simple Email Service API Reference](#).

### Increasing Your Amazon SES Sending Limits

When your account is out of the sandbox, your sending limits will increase if you are sending high-quality content and we detect that your utilization is approaching your current limits. Often, the system automatically increases your quota before you actually need it, and no further action is needed.

If your existing quota is not adequate for your needs and the system has not automatically increased your quota, you can open an [SES Sending Limits Increase case](#) in Support Center.

For a list of the information that you need when you open the case, see [Opening an SES Sending Limits Increase Case](#).

**Important**
Plan ahead. Be aware of your sending limits and try to stay within them. If you anticipate needing a higher quota than the system has allocated automatically, open an SES Sending Limits Increase case well before the date that you need the higher quota.

**Important**
If you anticipate needing to send more than one million emails per day, you must open an SES Sending Limits Increase case.

For Amazon SES to increase your quota, use the following guidelines:

- **Send high-quality content**—Send content that recipients want and expect.
- **Send real production content**—Send your actual production email. This enables Amazon SES to accurately evaluate your sending patterns, and verify that you are sending high-quality content.
• **Send near your current quota**—If your volume stays close to your quota without exceeding it, Amazon SES can detect this usage pattern and automatically increase your quota.

• **Have low bounce and complaint rates**—Try to minimize the numbers of bounces and complaints. Having high numbers of bounces and complaints can adversely affect your sending limits.

**Important**
Test emails that you send to your own email addresses may adversely affect your bounce and complaint metrics, or appear as low-quality content to our filters. Whenever possible, use the Amazon SES mailbox simulator to test your system. Emails that are sent to the mailbox simulator do not count toward your sending metrics or your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 162).

For information about opening an SES Sending Limits Increase case, see Opening an SES Sending Limits Increase Case (p. 130).

### Opening an SES Sending Limits Increase Case

To apply for higher sending limits for Amazon SES, open a case in Support Center by using the following instructions.

#### To request higher sending limits

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. Open an SES Sending Limits Increase case in Support Center. You can also access this link using the Dedicated IPs page in the Amazon SES console.
3. In the form, provide the following information:

   • **Region** – Choose the AWS Region for which you are requesting a sending limit increase. Your Amazon SES sandbox status and sending limits are separate for each AWS Region. For more information, see Regions and Amazon SES (p. 409).

   • **Limit** – Choose **Desired Daily Sending Quota** or **Desired Maximum Send Rate**. Sending limits are described in Managing Your Amazon SES Sending Limits (p. 127).

   **Note**
   The rate at which Amazon SES accepts your messages might be less than the maximum send rate.

   • **New limit value** – Enter the amount you are requesting. Only request the amount that you think you’ll need. Remember that you are not guaranteed to receive the amount you request, and the higher the limit you request, the more justification you need to be considered for that amount.

   • **Mail Type** – Choose the type of email you plan to send using your dedicated IP address. If multiple values apply, choose the type that will make up the majority of your email sending.

   • **Website URL** – Type the URL of your website. Providing this information will help us better understand the type of content you plan to send.

   • **My email sending complies with the AWS Service Terms and AUP** – Choose the option that applies to your use case.

   • **I only send to recipients who have specifically requested my mail** – Choose the option that applies to your use case.

   • **I have a process to handle bounces and complaints** – Choose the option that applies to your use case.

   • **Use Case Description** – Describe the ways in which you will use Amazon SES to send email. To help us process your request more quickly, please answer the following questions:

     • How will you build or acquire your mailing list?
     • How will you handle bounces and complaints?
What Happens When You Reach Your Sending Limits

- How can recipients unsubscribe from your mailing list, and how will you respond to those requests?
- How did you choose the increase amount you specified in this request?

When you finish, choose Submit. We will respond to the case after reviewing your request. Please allow one business day for processing.

What Happens When You Reach Your Amazon SES Sending Limits

If you attempt to send an email after you have reached your sending quota or maximum send rate, you will encounter a throttling error and the email will be dropped. The way you observe the error depends on whether you are calling the Amazon SES API directly, or accessing Amazon SES through the SMTP interface. The following sections describe both scenarios.

Note
You can send an email to multiple recipients as long as you have at least one email left before you reach your sending rate limit. Then, you have to wait until you build up the corresponding amount of sending rate quota before you can send again. For example, if your sending rate limit is one email per second, then you can send an email with 10 recipients once every 10 seconds. However, we do not recommend that you send an email to multiple recipients in one call to SendEmail.

For a technique to use when you reach your maximum send rate, see How to handle a "Throttling – Maximum sending rate exceeded" error on the Amazon SES blog.

Reaching Sending Limits with the Amazon SES API

If your application attempts to send an email beyond your sending limits, the application will encounter a Throttling error with one of the following messages:

- Daily message quota exceeded
- Maximum sending rate exceeded

A throttling error might occur because of incorrect predictions of email volume, or bursts of transactional email that are higher than expected. To handle a throttling error, program your application to wait for a random interval of between 0 and 10 minutes, and then retry the send request.

Reaching Sending Limits with SMTP

If you reach your sending limits when you are accessing Amazon SES through the SMTP interface, your SMTP client will receive one of the following errors:

- 454 Throttling failure: Maximum sending rate exceeded
- 454 Throttling failure: Daily message quota exceeded

However, SMTP clients handle these errors in various ways. For example, if you use Microsoft Outlook as your email client, and you attempt to send past your sending quota, you get a Send/Receive error that displays the following text in the status pane at the bottom of the Outlook client window:

Task 'andrew@example.net- Sending' reported error (0x800CCC7F): 'Establishing an encrypted connection to your outgoing (SMTP) server failed. If this problem
continues, contact your server administrator or Internet service provider (ISP). The server responded: 454 Throttling failure: Daily message quota exceeded.’

The way in which these errors are handled depends on the SMTP client that you use; some SMTP clients may not display the error code at all.

Using Sending Authorization with Amazon SES

You can configure Amazon SES to authorize other users to send emails from addresses or domains that you own (your identities) using their own Amazon SES accounts. This feature, called sending authorization, lets you maintain control over your identities so that you can change or revoke the permissions at any time. For example, if you are a business owner, you can use sending authorization to enable a third party (such as an email marketing company) to send email from a domain you own.

If you want to authorize someone to send emails on your behalf, then you are an identity owner. If you are an identity owner, we recommend that you read the following sections:

- Overview of Sending Authorization (p. 132)
- Sending Authorization Policies (p. 134)
- Sending Authorization Policy Examples (p. 138)
- Identity Owner Tasks (p. 144)

If you have been authorized to send emails on behalf of someone else, then you are a delegate sender. If you are a delegate sender, we recommend that you read the following sections:

- Overview of Sending Authorization (p. 132)
- Delegate Sender Tasks (p. 149)

Note

You can also control access to Amazon SES by using IAM policies. IAM policies constrain what individual IAM users can do, while sending authorization policies constrain how individual verified identities can be used. Further, only sending authorization policies can grant cross-account access. For more information about using IAM policies with Amazon SES, see Controlling Access to Amazon SES (p. 354).

Overview of Amazon SES Sending Authorization

This topic provides an overview of the sending authorization process and then explains how Amazon SES email-sending features, such as sending limits and notifications, work with sending authorization.

This section uses the following terms:

- **Identity** – An email address or domain that Amazon SES users use to send email.
- **Identity owner** – An Amazon SES user who has verified ownership of an email address or domain by using the procedures described in Verifying Identities (p. 43).
- **Delegate sender** – An entity that is authorized to send email from an identity it does not own. An AWS account, an AWS Identity and Access Management (IAM) user, or an AWS service can have this cross-account authority.
- **Sending authorization policy** – A document that you attach to an identity to specify who may send for that identity and under which conditions.
• **Amazon Resource Name (ARN)** – A standardized way to uniquely identify an AWS resource across all AWS services. In the case of sending authorization, the resource is the identity that the identity owner wants the delegate sender to use. An example of an ARN is `arn:aws:ses:us-west-2:123456789012:identity/example.com`.

### Sending Authorization Process

Sending authorization is based on sending authorization policies. If you want to enable a delegate sender to send on your behalf, you create a sending authorization policy and associate the policy to your identity by using the Amazon SES console or the Amazon SES API. When the delegate sender attempts to send an email through Amazon SES on your behalf, the delegate sender passes the ARN of your identity in the request or in the header of the email.

When Amazon SES receives the request to send the email, it checks your identity's policy (if present) to determine if you have authorized the delegate sender to send on the identity's behalf. If the delegate sender is authorized, Amazon SES accepts the email; otherwise, Amazon SES returns an error message.

The following diagram shows the high-level relationship between sending authorization concepts:

The sending authorization process consists of the following steps:

1. The identity owner verifies an identity with Amazon SES by using the Amazon SES console or the Amazon SES API. For information about the verification procedure, see [Verifying Identities](p. 43).
2. The delegate sender gives the identity owner the AWS account ID, IAM user ARN, or AWS service name of the entity that will do the sending.
3. The identity owner creates a sending authorization policy and attaches the policy to the identity by using the Amazon SES console or the Amazon SES API.
4. The identity owner gives the delegate sender the ARN of the identity so that the delegate sender can provide the ARN to Amazon SES at the time of email sending.
5. The identity owner and delegate sender configure bounce, complaint, and delivery notifications separately by using either the Amazon SES console or the Amazon SES API. For information about setting up notifications, see [Monitoring Using Amazon SES Notifications](p. 226).
6. The delegate sender attempts to send an email through Amazon SES on behalf of the identity owner by passing the ARN of the identity owner's identity in the request or in the header of the email. The delegate sender can send the email by using either the Amazon SES SMTP interface or the Amazon SES API. Upon receiving the request, Amazon SES examines any policies that are attached to the
identity, and accepts the email if the delegate sender is authorized to use the specified "From" address and "Return Path" address; otherwise, Amazon SES returns an error and does not accept the message.

7. To deauthorize the delegate sender, the identity owner simply edits the sending authorization policy or deletes the policy entirely. Either action can be performed by using the Amazon SES console or the Amazon SES API.

For more information about how the identity owner or delegate sender perform those tasks, see Identity Owner Tasks (p. 144) or Delegate Sender Tasks (p. 149), respectively.

**Attribution of Email-Sending Features**

It is important to understand the role of the delegate sender and the identity owner with respect to Amazon SES email-sending features such as daily sending quota, bounces and complaints, DKIM signing, feedback forwarding, and so on. The attribution is the following:

- **Sending limits** – Email sent from the identity owner's identities are counted against the delegate sender's daily sending quota.
- **Bounces and complaints** – Bounce and complaint events are recorded in the delegate sender's Amazon SES account, and can therefore impact the delegate sender's reputation.
- **DKIM signing** – If the identity owner has enabled Easy DKIM signing for an identity, all email sent from that identity will be DKIM-signed, including email sent by a delegate sender. Only the identity owner has control of whether the emails are DKIM-signed.
- **Notifications** – Both the identity owner and the delegate sender can set up their own Amazon SNS notifications for bounces, complaints, and deliveries. However, feedback forwarding by email is only available to the identity owner.
- **Verification** – Identity owners are responsible for following the procedure in Verifying Identities (p. 43) to verify that they own the email addresses and domains that they are authorizing delegate senders to use. Delegate senders do not need to verify any email addresses or domains specifically for sending authorization.
- **AWS Regions** – The delegate sender must send the emails from the AWS Region in which the identity owner's identity is verified. The sending authorization policy that gives permission to the delegate sender must be attached to the identity in that region.

**Amazon SES Sending Authorization Policies**

To enable another AWS account, Identity Access and Management (IAM) user, or AWS service to send email through Amazon SES on your behalf, you create a **sending authorization policy**, which is a JSON document that you attach to an identity that you own. The policy explicitly lists who you are allowing to send for that identity, and under which conditions. All senders but you and the entities you explicitly grant permissions to in the policies are denied. An identity can have no policy, one policy, or multiple policies attached to it. You can also have one policy with multiple statements to achieve the effect of multiple policies.

Policies can be simple or can be configured to provide fine-grained control. For example, if you owned example.com, you could write a simple policy to grant AWS ID 123456789012 permission to send from that domain. A more detailed policy could specify that AWS ID 123456789012 can send email only from user@example.com and only within a specified date range.

Amazon SES sending authorization policies apply to email-sending APIs (SendEmail and SendRawEmail) only. They do not enable a user to access your AWS account in any other way.

**Note**
You cannot create Identity Policies that allow delegate senders to use the SendTemplatedEmail or SendBulkTemplatedEmail operations.
Policy Structure

Each sending authorization policy is a JSON document that is attached to an identity. Each policy includes the following sections:

- Policy-wide information at the top of the document.
- One or more individual statements, each of which describes a set of permissions.

Each statement includes the core information about a single permission. If a policy includes multiple statements, Amazon SES applies a logical OR across the statements at evaluation time. Similarly, if an identity has multiple policies attached to it, Amazon SES applies a logical OR across the policies at evaluation time.

The following example policy grants AWS account ID 123456789012 permission to send from the verified domain example.com.

```
{
    "Id":"ExampleAuthorizationPolicy",
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"AuthorizeAccount",
            "Effect":"Allow",
            "Principal":{
                "AWS":[
                    "123456789012"
                ]
            },
            "Action":[
                "SES:SendEmail",
                "SES:SendRawEmail"
            ]
        }
    ]
}
```

You can find more sending authorization policy examples at Sending Authorization Policy Examples (p. 138).

Policy Elements

This section describes the elements contained in sending authorization policies. First we describe policy-wide elements, and then we describe elements that apply only to the statement in which they are included. We follow with a discussion of how to add conditions to your statements.

For specific information about the syntax of the elements, see Grammar of the IAM Policy Language in the IAM User Guide.

Policy-Wide Information

There are two policy-wide elements: Id and Version. The following table provides information about these elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Uniquely identifies the policy.</td>
<td>No.</td>
<td>Any string</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
<td>Valid Values</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Version</td>
<td>Specifies the policy access language version.</td>
<td>No, but as a best practice, we recommend that you include this field with a value of &quot;2012-10-17&quot;.</td>
<td>Any string</td>
</tr>
</tbody>
</table>

**Statements Specific to the Policy**

Sending authorization policies require at least one statement. Each statement can include the elements described in the following table.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sid</td>
<td>Uniquely identifies the statement.</td>
<td>No.</td>
<td>Any string.</td>
</tr>
<tr>
<td>Effect</td>
<td>Specifies the result that you want the policy statement to return at evaluation time.</td>
<td>No, although a statement without an effect is useless.</td>
<td>&quot;Allow&quot; or &quot;Deny&quot;.</td>
</tr>
<tr>
<td>Resource</td>
<td>Specifies the identity to which the policy applies. This is the email address or domain that the identity owner is authorizing the delegate sender to use.</td>
<td>Yes.</td>
<td>An identity's ARN, as specified in the Amazon SES console.</td>
</tr>
<tr>
<td>Principal</td>
<td>Specifies the AWS account, IAM user, or AWS service that receives the permission in the statement.</td>
<td>Yes.</td>
<td>A valid AWS account ID, IAM user ARN, or AWS service. AWS account IDs and IAM user ARNs are specified using &quot;AWS&quot; (for example, &quot;AWS&quot;: [&quot;123456789012&quot;] or &quot;AWS&quot;: [&quot;arn:aws:iam::123456789012:root&quot;]). AWS service names are specified using &quot;Service&quot; (for example, &quot;Service&quot;: [&quot;cognito-idp.amazonaws.com&quot;]). For examples of the format of IAM user ARNs, see the AWS General Reference.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
<td>Valid Values</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Action</td>
<td>Specifies the email-sending action to which the statement applies.</td>
<td>Yes.</td>
<td>&quot;ses:SendEmail&quot;, &quot;ses:SendRawEmail&quot; (one or both).</td>
</tr>
<tr>
<td></td>
<td>If you use the custom policy editor, you can also set the action to &quot;ses:<em>&quot; to encompass both APIs. If your sender will access Amazon SES through the SMTP interface, you must at least specify &quot;ses:SendRawEmail&quot;, or use &quot;ses:</em>&quot;.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Condition</td>
<td>Specifies any restrictions or details about the permission.</td>
<td>No.</td>
<td>See the information about conditions following this table.</td>
</tr>
</tbody>
</table>

**Conditions**

A *condition* is any restriction about the permission in the statement. The part of the statement that specifies the conditions can be the most detailed of all the parts. A *key* is the specific characteristic that is the basis for access restriction, such as the date and time of the request.

You use both conditions and keys together to express the restriction. For example, if you want to restrict the delegate sender from making requests to Amazon SES on your behalf after July 30, 2015, you use the condition called DateLessThan. You use the key called aws:CurrentTime and set it to the value 2015-07-30T00:00:00Z.

You can use any of the AWS-wide keys listed at Available Keys in the IAM User Guide, or you can use one of the following keys specific to Amazon SES:

<table>
<thead>
<tr>
<th>Condition Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ses:Recipients</td>
<td>Restricts the recipient addresses, which include the To:, &quot;CC&quot;, and &quot;BCC&quot; addresses.</td>
</tr>
<tr>
<td>ses:FromAddress</td>
<td>Restricts the &quot;From&quot; address.</td>
</tr>
<tr>
<td>ses:FromDisplayName</td>
<td>Restricts the contents of the string that is used as the &quot;From&quot; display name (sometimes called &quot;friendly from&quot;). For example, the display name of &quot;John Doe <a href="mailto:johndoe@example.com">johndoe@example.com</a>&quot; is John Doe.</td>
</tr>
<tr>
<td>ses:FeedbackAddress</td>
<td>Restricts the &quot;Return Path&quot; address, which is the address where bounce and complaints can be sent to you by email feedback forwarding. For information about email feedback forwarding, see Amazon SES Notifications Through Email (p. 227).</td>
</tr>
</tbody>
</table>

It is common to use the StringEquals and StringLike conditions with the Amazon SES keys. These conditions are for case-sensitive string matching. For StringLike, the values can include a multi-
character match wildcard (*) or a single-character match wildcard (?) anywhere in the string. For example, the following condition specifies that the delegate sender can only send from a "From" address that starts with *invoicing* and ends with *example.com*:

```
"Condition": {
  "StringLike": {
    "ses:FromAddress": "invoicing+.example.com"
  }
}
```

**Note**
When you want to disallow access to an email address, use wildcards to ensure that you are completely preventing access to all forms of that address. For example, to disallow sending from *admin@example.com*, you can prevent access to alternatives such as *admin@example.com* and *admin+1@example.com* by specifying the following condition:

```
"Condition": {
  "StringNotLike": {
    "ses:FromAddress": "*admin*.example.com"
  }
}
```

For more information about how to specify conditions, see **Condition** in the *IAM User Guide*.

**Policy Requirements**

Each policy must adhere to the following requirements:

- Each policy must include at least one statement.
- Each policy must include at least one valid principal.
- Each policy must specify one resource, and that resource must be the ARN of the identity to which the policy is attached.
- Identity owners can associate up to 20 policies with each unique identity.
- Policies must not exceed 4 kilobytes (KB).
- Policy names cannot exceed 64 characters and can only include alphanumeric characters, dashes, and underscores.

**Amazon SES Sending Authorization Policy Examples**

Sending authorization enables you to specify the fine-grained conditions under which you allow delegate senders to send on your behalf. The following examples show you how to write policies to control different aspects of sending:

- Specifying the Delegate Sender (p. 139)
- Restricting the "From" Address (p. 140)
- Restricting the Destination of Bounce and Complaint Feedback (p. 140)
- Restricting the Time at which the Delegate can Send Email (p. 141)
- Restricting the Email-Sending Action (p. 142)
- Restricting the Display Name of the Email Sender (p. 142)
- Using Multiple Statements (p. 143)
Specifying the Delegate Sender

The principal, which is the entity to which you are granting permission, can be an AWS account, an AWS Identity and Access Management (IAM) user, or an AWS service.

The following example shows a simple policy that allows AWS ID 123456789012 to send email from the verified identity example.com (which is owned by AWS account 888888888888). The Condition statement in this policy only allows the delegate (that is, AWS ID 123456789012) to send email from the address marketing+.*/example.com, where .* is any string that the sender wants to add after marketing+.

```json
{
   "Id":"SampleAuthorizationPolicy",
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid":"AuthorizeMarketer",
         "Effect":"Allow",
         "Principal": {
            "AWS": ["123456789012"]
         },
         "Action": ["SES:SendEmail", "SES:SendRawEmail"],
         "Condition": {
            "StringLike": {
               "ses:FromAddress":"marketing+.*@example.com"
            }
         }
      }
   ]
}
```

The following example policy grants permission to two IAM users to send from identity example.com. IAM users are specified by their Amazon Resource Name (ARN).

```json
{
   "Id":"ExampleAuthorizationPolicy",
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid":"AuthorizeIAMUser",
         "Effect":"Allow",
         "Principal": {
            "AWS": ["arn:aws:iam::111122223333:user/John",
                     "arn:aws:iam::444455556666:user/Jane"],
         },
         "Action": ["SES:SendEmail", "SES:SendRawEmail"],
      }
   ]
}
```

The following example policy grants permission to Amazon Cognito to send from identity example.com.
Sending Authorization Policy Examples

```json
{
  "Id":"ExampleAuthorizationPolicy",
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"AuthorizeService",
      "Effect":"Allow",
      "Principal":{
        "Service":[
          "cognito-idp.amazonaws.com"
        ],
    },
    "Action":[
      "SES:SendEmail",
      "SES:SendRawEmail"
    ]
  ]
}
```

Restricting the "From" Address

If you use a verified domain, you may want to create a policy that only allows the delegate sender to send from a specified email address. To restrict the "From" address, you set a condition on the key called `ses:FromAddress`. The following policy enables AWS account ID 123456789012 to send from the identity `example.com`, but only from the email address `sender@example.com`.

```json
{
  "Id":"ExamplePolicy",
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"AuthorizeFromAddress",
      "Effect":"Allow",
      "Principal":{
        "AWS":[
          "123456789012"
        ],
    },
    "Action":[
      "SES:SendEmail",
      "SES:SendRawEmail"
    ],
    "Condition":{
      "StringEquals":{
        "ses:FromAddress":"sender@example.com"
      }
    }
  ]
}
```

Restricting the Destination of Bounce and Complaint Feedback

If a delegate sender is sending on your behalf and you want to ensure that bounce and complaint notifications are forwarded to you by email, you need to do two things: you must enable email feedback forwarding for the identity by using the procedure in Amazon SES Notifications Through Email (p. 227), and you must restrict the "Return Path" of the emails to an email address that you own by setting a condition on the `ses:FeedbackAddress` key.
The following sending authorization policy enables AWS account ID 123456789012 to send from the identity example.com as long as the "Return Path" of the email is set to feedback@example.com.

```
{
    "Id": "ExamplePolicy",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ControlReturnPath",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "123456789012"
                ],
            },
            "Action": ["SES:SendEmail", "SES:SendRawEmail"],
            "Condition": {
                "StringEquals": {
                    "ses:FeedbackAddress": "feedback@example.com"
                }
            }
        }
    ]
}
```

Restricting the Time at which the Delegate can Send Email

You can also configure your sender authorization policy so that a delegate sender can only send email at a certain time of day, or within a certain date range. For example, if you plan to send an email campaign during the month of September 2018, you can use the following policy to limit the delegate's ability to send email to that month only.

```
{
    "Id": "ExamplePolicy",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ControlTimePeriod",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "123456789012"
                ],
            },
            "Action": ["SES:SendEmail", "SES:SendRawEmail"],
            "Condition": {
                "DateGreaterThan": {
                    "aws:CurrentTime": "2018-08-31T12:00Z"
                },
                "DateLessThan": {
                    "aws:CurrentTime": "2018-10-01T12:00Z"
                }
            }
        }
    ]
}
```
Restricting the Email-Sending Action

There are two actions that senders can use to send an email with Amazon SES: SendEmail and SendRawEmail, depending on how much control the sender wants over the format of the email. Sending authorization policies enable you to restrict the delegate sender to one of those two actions. However, many identity owners leave the details of the email-sending calls up to the delegate sender by enabling both actions in their policies.

**Note**

If you want to enable the delegate sender to access Amazon SES through the SMTP interface, you must choose `SendRawEmail` at a minimum.

If your use case is such that you want to restrict the action, you can do so by including only one of the actions in your sending authorization policy. The following example shows you how to restrict the action to `SendRawEmail`.

```
{
  "Id":"ExamplePolicy",
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"ControlAction",
      "Effect":"Allow",
      "Principal":{
        "AWS":[
          "123456789012"
        ]
      },
      "Action":[
        "SES:SendRawEmail"
      ]
    }
  ]
}
```

Restricting the Display Name of the Email Sender

Some email clients display the "friendly" name of the email sender (if the email header provides it), rather than the actual "From" address. For example, the display name of "John Doe <johndoe@example.com>" is John Doe. For instance, you might send emails from `user@example.com`, but you prefer that recipients see that the email is from Marketing rather than from `user@example.com`. The following policy enables AWS account ID 123456789012 to send from identity `example.com`, but only if the display name of the "From" address includes Marketing.

```
{
  "Id":"ExamplePolicy",
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"AuthorizeFromAddress",
      "Effect":"Allow",
      "Principal":{
        "AWS":[
          "123456789012"
        ]
      },
      "Action":[
        "SES:SendRawEmail"
      ]
    }
  ]
}
```
Using Multiple Statements

Your sending authorization policy can include multiple statements. The following example policy has two statements. The first statement authorizes two AWS accounts to send from sender@example.com as long as the "From" address and the feedback address both use the domain example.com. The second statement authorizes an IAM user to send email from sender@example.com as long as the recipient's email address is under the example.com domain.

```json
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AuthorizeAWS",
         "Effect":"Allow",
         "Resource":"arn:aws:ses:us-east-1:999999999999:identity/sender@example.com",
         "Principal":{
            "AWS":[
               "111111111111",
               "222222222222"
            ],
         },
         "Action":[
            "SES:SendEmail",
            "SES:SendRawEmail"
         ],
         "Condition":{
            "StringLike":{
               "ses:FromAddress"":"*@example.com",
               "ses:FeedbackAddress":"*@example.com"
            }
         }
      },
      {
         "Sid":"AuthorizeInternal",
         "Effect":"Allow",
         "Resource":"arn:aws:ses:us-east-1:999999999999:identity/sender@example.com",
         "Principal":{
            "AWS":"arn:aws:iam::333333333333:user/Jane"
         },
         "Action":[
            "SES:SendEmail",
            "SES:SendRawEmail"
         ],
         "Condition":{
            "ForAllValues:StringLike":{
               "ses:Recipients":"*@example.com"
            }
         }
      }
   ]
}
```
Identity Owner Tasks for Amazon SES Sending Authorization

This section describes the steps that identity owners must take when configuring sending authorization. As an identity owner, you must complete the following tasks:

- Verifying an Identity (p. 144)
- Setting Up Notifications (p. 144)
- Getting Information from the Delegate Sender (p. 144)
- Creating a Policy (p. 145)
- Providing the Delegate Sender with the Identity Information (p. 148)
- Managing Your Policies (p. 148)

Verifying an Identity for Amazon SES Sending Authorization

The first step in configuring sending authorization is to prove that you own the email address or domain from which the delegate sender will send email. The verification procedure is described in Verifying Identities (p. 43).

You can confirm that an email address or domain is verified by checking its status in the Identity Management section of the Amazon SES console or by using the GetIdentityVerificationAttributes API operation.

Setting Up Identity Owner Notifications for Amazon SES Sending Authorization

When a delegate sender sends email using an identity you own, bounces and complaints that those emails generate count toward the delegate sender's bounce and complaint metrics rather than your own. However, if you want to be informed of these email-sending outcomes, you can set up notifications by email or by Amazon SNS notifications just as you would for any other identity. For more information about receiving bounce and complaint notifications, see Monitoring Using Amazon SES Notifications (p. 226).

Delegate senders can set up their own Amazon SNS notifications for the identities that you have authorized them to use, although only you, the identity owner, can control whether feedback can be directly forwarded to your identity via email.

Getting Information from the Delegate Sender for Amazon SES Sending Authorization

Your sending authorization policy must specify at least one principal, which is the entity to which you are granting access. For Amazon SES sending authorization policies, the principal can be an AWS account, an AWS Identity and Access Management (IAM) user, or an AWS service.

The type of principal you choose depends on your preference, but if you want the finest grain control, ask the delegate sender to set up an IAM user so that only one delegate sender can send for you rather
than any user in the delegate sender’s AWS account. The delegate sender can find information about setting up an IAM user in Creating an IAM User in Your AWS Account in the IAM User Guide.

After you have decided whether you want to grant access to an AWS account, an IAM user, or an AWS service, ask the delegate sender for the AWS account ID or the IAM user’s Amazon Resource Name (ARN) so that you can include it in your sending authorization policy. You can refer your delegate sender to the instructions for finding this information in Providing Information to the Identity Owner (p. 150). If the delegate sender is an AWS service, see the documentation for that service to determine the service name.

Creating a Policy for Amazon SES Sending Authorization

To authorize a delegate sender to send emails for one of your identities, you create a sending authorization policy, and then attach that policy to the identity. Identities can have zero policies, one policy, or multiple policies. However, each policy may only be associated with one identity.

Important
Policies attached to email address identities override policies attached to the corresponding domain identities. For example, say that you have verified example.com and user@example.com. If you create a policy for example.com that disallows a delegate sender, and you create a policy for user@example.com that allows that delegate sender, the delegate sender will be able to send from user@example.com if they specify the ARN of user@example.com in the request to send the email.

You can create a sending authorization policy in the following ways:

- **By using the Policy Generator** – You can create a simple policy by using the Policy Generator in the Amazon SES console. In addition to specifying who can send the emails, you can constrain the email-sending with conditions based on the time and date range in which emails can be sent, the “From” address, the “From” display name, the address to which bounces and complaints are sent, the recipient addresses, and the source IP. You might also want to use the Policy Generator to create the structure of a simple policy and then customize it later by editing the policy.

- **By creating a Custom Policy** – If you want to include more advanced conditions or use an AWS service as the principal, you can create a custom policy and attach it to the identity by using the Amazon SES console or the Amazon SES API.

This topic describes both methods.

Using the Policy Generator

You can use the Policy Generator to create a simple authorization policy by using the following procedure.

To create a policy by using the Policy Generator

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Identity Management, choose either Domains or Email Addresses.
3. In the list of identities, choose the identity for which you want to create a policy.
4. In the details pane, expand Identity Policies, choose Create Policy, and then choose Policy Generator.
5. In the wizard, create a policy statement by choosing values for the following fields. You can find information about these options in Sending Authorization Policies (p. 134).
   - **Effect** – If you want to grant access, choose Allow; otherwise, choose Deny.
• **Principals** – Enter either the 12-digit AWS account ID or the ARN of an IAM user that you are allowing or denying access, and then choose **Add**. You can add more principals by repeating this step. An example of an AWS account ID is 123456789012 and an example of an IAM user ARN is `arn:aws:iam::123456789012:user/John`.

**Note**
The policy generator wizard does not currently support AWS service principals. To add an AWS service principal, you must either [create a custom policy](#) or use the policy generator to add an AWS account or IAM user principal, and then [edit](#) the policy.

• **Actions** – Choose the email-sending access to which this policy applies. Typically, identity owners choose both options to give the delegate sender the freedom to choose how to implement the email sending. For more information, see [Statements Specific to the Policy](#).

6. (Optional) If you want to add restrictions to the policy, choose **Add Conditions**, and then choose the following information:

• **Key** – This is the characteristic that is the basis for access restriction. The Policy Generator lets you choose an Amazon SES-specific key or one of a few commonly used AWS-wide keys (current time and source IP). For details, see [Conditions](#). If you want to specify the more advanced AWS-wide keys listed in [Available Keys](#), you can edit the policy after you create it.

• **Condition** – This is the type of condition that you want to specify. For example, there are string conditions, numeric conditions, date and time conditions, and so on. For a list of conditions, see [Condition Types](#) in the [IAM User Guide](#).

• **Value** – This is the value that will be tested against the condition. For examples, see the policies in [Sending Authorization Policy Examples](#).

After you choose the key, condition, and value, choose **Add Condition**. The condition appears in the **Conditions** list. You can remove conditions by choosing **Remove** next to a condition in the list. You can add another condition by choosing **Add Conditions** again.
7. When you finish adding conditions, choose Add Statement. The statement appears in the Statements list, where you can choose to edit or remove it. You can add additional statements by repeating steps 5–7.

8. When you finish adding statements, choose Next.

9. In the Edit Policy dialog box, review your policy, edit it if necessary, and then choose Apply Policy.

Creating a Custom Policy

If you want to create a custom policy and attach it to an identity, you have the following options:

- **Using the Amazon SES API** – Create a policy in a text editor and then attach the policy to the identity by using the PutIdentityPolicy API described in the Amazon Simple Email Service API Reference.
- **Using the Amazon SES console** – Create a policy in a text editor and attach it to an identity by pasting it into the Custom Policy editor in the Amazon SES console. The following procedure describes this method.

To create a custom policy by using the Custom Policy editor

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under **Identity Management**, choose either **Domains** or **Email Addresses**.
3. In the list of identities, choose the identity for which you want to create a policy.
4. In the details pane, expand **Identity Policies**, choose **Create Policy**, and then choose **Custom Policy**.
5. In the **Edit Policy** pane, paste the text of your policy.
6. Choose **Apply Policy**.

### Providing the Delegate Sender with the Identity Information for Amazon SES Sending Authorization

After you create your sending authorization policy and attach it to your identity, you can provide the delegate sender with the Amazon Resource Name (ARN) of the identity. The delegate sender will pass that ARN to Amazon SES in the email-sending operation or in the header of the email. Use the following procedure to find your identity's ARN.

#### To find the ARN of an identity

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under **Identity Management**, choose either **Domains** or **Email Addresses**.
3. In the list of identities, choose the identity to which you attached the sending authorization policy.
4. At the top of the details pane, after **Identity ARN**, you will see the identity's ARN. It will look similar to `arn:aws:ses:us-east-1:123456789012:identity/user@example.com`. Copy the entire ARN and give it to your delegate sender.

### Managing Your Policies for Amazon SES Sending Authorization

In addition to creating and attaching policies to identities as explained in Creating a Policy (p. 145), you can edit, remove, list, and retrieve an identity's policies, as described in the following sections.

#### Note

To revoke permissions, you can either edit a policy or remove it.

### Editing a Policy

The easiest way to edit a policy is to use the Amazon SES console. If you want to use the Amazon SES API instead, you can use the `GetIdentityPolicies` operation to retrieve the policy, edit the policy using a text editor, and then use the `PutIdentityPolicy` operation to overwrite the older policy.

The following procedure shows you how to edit a policy by using the Amazon SES console.

#### To edit a policy by using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under **Identity Management**, choose either **Domains** or **Email Addresses**.
3. In the list of identities, choose the identity that is associated with the policy that you want to edit.
4. In the details pane, expand **Identity Policies**.
5. Next to the policy that you want to edit, choose **Edit Policy**.
6. In the Edit Policy pane, edit the policy, and then choose Apply Policy.
7. In the Overwrite Existing Policy dialog box, choose Overwrite.

Removing a Policy

To revoke permissions at any time, you can simply remove the policy. You can remove a policy by using the DeleteIdentityPolicy API operation, or you can use the Amazon SES console, as described in the following procedure.

Important
After you remove a policy, there is no way to get it back. We recommend that you back up the policy by copying and pasting it into a text file before you remove the policy.

To remove a policy by using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Identity Management, choose either Domains or Email Addresses.
3. In the list of identities, choose the identity that is associated with the policy that you want to remove.
4. In the details pane, expand Identity Policies. Next to the policy that you want to remove, choose Remove Policy.
5. In the Remove Policy dialog box, choose Yes, Remove Policy.

Listing and Retrieving Policies

You can list the policies that are attached to an identity by using the ListIdentityPolicies API operation. You can also retrieve the policies themselves by using the GetIdentityPolicies API operation.

You can also use the Amazon SES console to perform both of these tasks, as described in the following procedure.

To list and show the policies attached to an identity by using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Identity Management, choose either Domains or Email Addresses.
3. In the list of identities, choose the identity for which you want to see policies.
4. In the details pane, expand Identity Policies.
5. Next to the policy that you want to view, choose Show Policy.

Delegate Sender Tasks for Amazon SES Sending Authorization

As a delegate sender, you are sending cross-account emails. This means that you are sending emails on behalf of an identity that you do not own, but are authorized to use. Even though you are sending on the identity owner's behalf, bounces and complaints count toward the bounce and complaint metrics for your AWS account, and the number of messages you send counts toward your sending quota. You
are also responsible for requesting any sending limit increases that you might need in order to send the identity owner's emails.

As a delegate sender, you must complete the following tasks:

- Providing Information to the Identity Owner (p. 150)
- Using Delegate Sender Notifications (p. 150)
- Sending Emails for the Identity Owner (p. 153)

Providing Information to the Identity Owner for Amazon SES Sending Authorization

As a delegate sender, you must provide the identity owner with your AWS account ID or the Amazon Resource Name (ARN) of the AWS Identity and Access Management (IAM) user who will send email on behalf of the identity owner. You can find this information by using the following procedures.

To find your AWS account ID

2. In the navigation menu, choose your name, and then choose My Account.
3. Expand Account Settings. Your AWS account ID is listed at the top of this section.

To find the ARN of an IAM user

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose Users.
3. In the list of users, choose the user name. The Summary section displays the ARN. The ARN will resemble the following example: arn:aws:iam::123456789012:user/John.

Using Delegate Sender Notifications for Amazon SES Sending Authorization

As a delegate sender, you can set up Amazon Simple Notification Service (Amazon SNS) notifications to inform you of bounces, complaints, and deliveries. The format and content of these notifications are described in Amazon SES Notifications Through Amazon SNS (p. 229). Only the identity owner has the option to receive notifications by email feedback forwarding as described in Amazon SES Notifications Through Email (p. 227).

Important

As the delegate sender, bounces and complaints count toward your bounce and complaint metrics, not those of the identity owner. High bounce and complaint rates put your account at risk of being shut down, so ensure that you set up notifications and have a process in place to monitor the notifications and remove recipient addresses that have bounced or complained from your mailing list.

You will be charged standard Amazon SNS rates for bounce, complaint, and delivery notifications. For more information, see the Amazon SNS pricing page.

The following sections show you how to manage cross-account identity notifications:

- Setting Up an Amazon SES Cross-Account Identity Notification Configuration (p. 151)
Setting Up an Amazon SES Cross-Account Identity Notification Configuration

Before you set up notifications, you need to know the Amazon Resource Name (ARN) of the identity that the identity owner has authorized you to use, and for which you want to configure notifications. For example, the ARN for identity user@example.com would look similar to arn:aws:ses:us-east-1:123456789012:identity/user@example.com. If the identity owner has not given you the identity’s ARN, refer them to the procedure in Providing the Delegate Sender with the Identity Information (p. 148).

The easiest way to configure notifications is to use the Amazon SES console. You can also use the SetIdentityNotificationTopic API operation, passing the identity's ARN as the Identity parameter.

The following procedure shows you how to set up notifications by using the Amazon SES console.

**To set up Amazon SNS bounce, complaint, or delivery notifications by using the Amazon SES console**

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose Cross-Account Notifications.
3. Choose Add Notification Config.
4. In the Edit Notification Configuration dialog box, enter the ARN of the identity that the identity owner has authorized you to use, and for which you want to configure notifications. The identity cannot belong to the account that is currently logged in. If you want to configure notifications for your own identities, see Configuring Amazon SNS Notifications for Amazon SES (p. 229).
5. Specify the existing Amazon SNS topics that you want to use for bounces, complaints, or deliveries, or create a new Amazon SNS topic.
   - **Important**
     - The Amazon SNS topics that you use for Amazon SES notifications must be within the same AWS Region in which you are using Amazon SES.

     You can choose to publish bounce, complaint, and delivery notifications to the same Amazon SNS topic or to different Amazon SNS topics. If you want to use an Amazon SNS topic that you do not own, then the owner of that topic must configure an Amazon SNS access policy that allows your account to call the SNS:Publish action on their topic. For information about how to control access to your Amazon SNS topic through the use of IAM policies, see Managing Access to Your Amazon SNS Topics.
6. Choose Save Config to save your notification configuration. There may be a brief delay before these changes take effect.

   After you have configured your settings, you will receive bounce, complaint, or delivery notifications for the Amazon SNS topic. These notifications follow the structure described in Amazon SNS Notification Contents for Amazon SES (p. 231).

Editing an Amazon SES Cross-Account Notification Configuration

The easiest way to edit notification configurations is to use the Amazon SES console. If you want to use the Amazon SES API instead, you can use the SetIdentityNotificationTopic API operation and pass the identity's ARN as the Identity parameter.
The following procedure shows you how to edit a cross-account notification configuration by using the Amazon SES console.

**To edit a cross-account notification configuration by using the Amazon SES console**

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose **Cross-Account Notifications**.
   
   The cross-account identities for which you have set up notifications are listed in the **Cross-Account Notifications** details pane.
3. Choose the ARN of the identity for which you want to view the notification configuration.
4. Edit the notification settings, and then choose **Save Config**.

   **Note**
   Setting all notifications to **No SNS Topic** is the equivalent of removing the identity's notification configuration entirely. In this case, the ARN of the cross-account identity is removed from the list of cross-account identity ARNs in the Amazon SES console. This does not mean that you cannot continue to send for that identity; it just means that you are no longer set up to receive bounce, complaint, or delivery notifications for it. If you want to re-enable notifications, you must repeat the notification setup procedure described in Setting Up a Notification Configuration (p. 151).

**Viewing Your Amazon SES Cross-Account Identity Notifications**

The easiest way to view your notification configurations is to use the Amazon SES console. You can also use the `GetIdentityNotificationAttributes` API operation, passing the identity's ARN as the `Identity` parameter.

**Note**
The only cross-account identities displayed in the cross-account identity list are the identities for which you have configured notifications by using the procedure described in Setting Up a Notification Configuration (p. 151).

**To view your cross-account notification configurations by using the Amazon SES console**

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose **Cross-Account Notifications**.
   
   The cross-account identities for which you have set up notifications are listed in the **Cross-Account Notifications** details pane.
3. Choose the ARN of an identity.
   
   The **Edit Configuration Notification** dialog box displays the identity's settings.

**Removing an Amazon SES Cross-Account Identity Notification Configuration**

The easiest way to remove a notification configuration is to use the Amazon SES console. You can also use the `SetIdentityNotificationTopic` API operation, passing the identity's ARN as the `Identity` parameter, and passing null for the `SnsTopic` parameter. To completely remove the notification configuration, you must perform this operation for each type of notification type (bounce, complaint, or delivery) that was set.

**Note**
When you remove a notification configuration, the ARN of the cross-account identity is removed from the list of cross-account identity ARNs in the Amazon SES console. This does not mean
that you cannot continue to send for that identity; it just means that you are no longer set up to receive bounce, complaint, or delivery notifications for it. If you want to re-enable notifications, you need to repeat the notification setup procedure described in Setting Up a Notification Configuration (p. 151).

The following procedure shows you how to remove a cross-account notification configuration by using the Amazon SES console.

**To remove a cross-account notification configuration by using the Amazon SES console**

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose **Cross-Account Notifications**.
   
   The cross-account identities for which you have set up notifications are listed in the **Cross-Account Notifications** details pane.
3. Choose the box to the left of the cross-identity that you want to remove, and then choose **Remove**.
4. In the **Remove Cross-Account Notification Config** dialog box, choose **Delete Notification config**.

   The ARN of the cross-account identity no longer appears in the list of cross-account identity ARNs. This does not mean that you cannot send for the identity, just that you no longer have configured notifications for it.

**Sending Emails for the Identity Owner for Amazon SES Sending Authorization**

As a delegate sender, you send emails the same way that other Amazon SES senders do, except that you provide the ARN of the identity that the identity owner has authorized you to use. When you call Amazon SES to send the email, Amazon SES checks to see if the identity that you specified has a policy that authorizes you to send for it.

There are different ways that you can specify the identity’s ARN when you send an email. The method that you can use depends on whether you send the email by using the Amazon SES API operations or the Amazon SES SMTP interface.

**Important**

To successfully send an email from an identity owner's identity, you must connect to the Amazon SES endpoint of the AWS Region in which the identity is verified. The sending authorization policy that grants you permission must be attached to the identity in that region.

**Using the Amazon SES API**

As with any Amazon SES email sender, if you access Amazon SES through the Amazon SES API (either directly through HTTPS or indirectly through an AWS SDK), you can choose between one of two email-sending actions: **SendEmail** and **SendRawEmail**. The Amazon Simple Email Service API Reference describes the details of these APIs, but we provide an overview of the sending authorization parameters here.

**SendRawEmail**

If you want to use **SendRawEmail** so that you can control the format of your emails, you can specify the cross-account identity in one of two ways:

- **Pass optional parameters to the SendRawEmail API**. The required parameters are described in the following table:
Delegate Sender Tasks

### Parameter | Description
---|---
SourceArn | The ARN of the identity that is associated with the sending authorization policy that permits you to send for the email address specified in the Source parameter of SendRawEmail.

**Note**
For most use cases, we recommend that you specify the SourceArn and do not specify either the FromArn or ReturnPathArn. If you only specify the SourceArn, Amazon SES sets the "From" address and the "Return Path" addresses to the identity specified in SourceArn.

FromArn | The ARN of the identity that is associated with the sending authorization policy that permits you to specify a particular "From" address in the header of the raw email.

ReturnPathArn | The ARN of the identity that is associated with the sending authorization policy that permits you to use the email address specified in the ReturnPath parameter of SendRawEmail.

- **Include X-headers in the email.** X-headers are custom headers that you can use in addition to standard email headers (such as the From, Reply-To, or Subject headers). Amazon SES recognizes three X-headers that you can use to specify sending authorization parameters:

  **Important**
  Do not include these X-headers in the DKIM signature, because they are removed by Amazon SES before sending the email.

| X-Header | Description |
---|---|
X-SES-SOURCE-ARN | Corresponds to the SourceArn. |
X-SES-FROM-ARN | Corresponds to the FromArn. |
X-SES-RETURN-PATH-ARN | Corresponds to the ReturnPathArn. |

Amazon SES removes all X-headers from the email before sending it. If you include multiple instances of an X-header, Amazon SES only uses the first instance.

The following example shows an email that includes sending authorization X-headers:

```

From: sender@example.com
To: recipient@example.com
Return-Path: feedback@example.com
Subject: subject
Content-Type: multipart/alternative;
  boundary="----=_boundary"
```
If you use the \texttt{SendEmail} operation, you can specify the cross-account identity by passing in the optional parameters below. You cannot use the X-header method when you use the \texttt{SendEmail} operation.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{SourceArn}</td>
<td>The ARN of the identity that is associated with the sending authorization policy that permits you to send for the email address specified in the \texttt{Source} parameter of \texttt{SendRawEmail}.</td>
</tr>
<tr>
<td>\texttt{ReturnPathArn}</td>
<td>The ARN of the identity that is associated with the sending authorization policy that permits you to use the email address specified in the \texttt{ReturnPath} parameter of \texttt{SendRawEmail}.</td>
</tr>
</tbody>
</table>

The following example shows how to send an email that includes the \texttt{SourceArn} and \texttt{ReturnPathArn} attributes using the \texttt{SendEmail} operation and the \texttt{SDK for Python}.

```python
import boto3
from botocore.exceptions import ClientError

# Create a new SES resource and specify a region.
client = boto3.client('ses', region_name="us-west-2")

# Try to send the email.
try:
    # Provide the contents of the email.
    response = client.send_email(
        Destination={
            'ToAddresses': [
                'recipient@example.com',
            ],
        },
        Message={
            'Body': {
                'Html': {
                    'Charset': 'UTF-8',
                    'Data': 'This email was sent with Amazon SES.',
                },
            },
            'Subject': {
                'Charset': 'UTF-8',
                'Data': 'Amazon SES Test',
            },
        },
    )
```
Using the Amazon SES SMTP interface

If you are using the Amazon SES SMTP interface for cross-account sending, the only method you can use is to include the X-headers as `SendRawEmail` described earlier.

Using Dedicated IP Addresses with Amazon SES

When you create a new Amazon SES account, your emails are sent from IP addresses that are shared with other Amazon SES users. For an additional monthly charge, you can lease dedicated IP addresses that are reserved for your exclusive use. Both of these options offer unique benefits and drawbacks, which are summarized in the following table; click an item in the **Benefit** column for additional information about that benefit.

<table>
<thead>
<tr>
<th>Benefit</th>
<th>Shared IP addresses</th>
<th>Dedicated IP addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ready to use with no additional setup (p. 157)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Reputation managed by AWS (p. 157)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Good for customers with continuous, predictable sending patterns (p. 157)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Good for customers with less predictable sending patterns (p. 157)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Good for high-volume senders (p. 157)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Good for low-volume senders (p. 157)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Additional monthly costs (p. 158)</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Complete control over sender reputation (p. 158)</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Isolate reputation by email type, recipient, or other factors (p. 158)</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Ease of Setup

If you choose to use shared IP addresses, then you do not need to perform any additional configuration; your Amazon SES account is ready to send email as soon as you verify an email address and move out of the sandbox.

If you choose to lease dedicated IP addresses, you must determine how many dedicated IP addresses you need, submit a request, and optionally set up dedicated IP pools.

Reputation Managed by AWS

IP address reputations are based largely on historical sending patterns and volume. An IP address that sends consistent volumes of email over a long period of time usually has a good reputation.

Shared IP addresses are used by several Amazon SES customers. Together, these customers send a large volume of email. AWS carefully manages this outbound traffic in order to maximize the reputations of the shared IP addresses.

If you use dedicated IP addresses, it is your responsibility to maintain your sender reputation by sending consistent and predictable volumes of email.

Predictability of Sending Patterns

An IP address with a consistent history of sending email has a better reputation than one that suddenly starts sending out large volumes of email with no prior sending history.

If your email sending patterns are irregular—that is, they do not follow a predictable pattern—then shared IP addresses are probably a better fit for your needs. When you use shared IP addresses, you can increase or decrease your email sending patterns as the situation demands.

If you use dedicated IP addresses, you must warm up those addresses by sending an amount of email that gradually increases every day. The process of warming up new IP addresses is described in Warming up Dedicated IP Addresses (p. 160). Once your dedicated IP addresses are warmed up, you must then maintain a consistent sending pattern.

Volume of Outbound Email

Dedicated IP addresses are best suited for customers who send large volumes of email. Most internet service providers (ISPs) only track the reputation of a given IP address if they receive a significant volume of mail from that address. For each ISP with which you want to cultivate a reputation, you should send several hundred emails within a 24-hour period at least once per month.

In some cases, you may be able to use dedicated IP addresses if you do not send large volumes of email. For example, dedicated IP addresses may work well if you send to a small, well-defined group of
recipients whose mail servers accept or reject email using a list of specific IP addresses, rather than IP address reputation.

**Additional Costs**

The use of shared IP addresses is included in the standard Amazon SES pricing. Leasing dedicated IP addresses incurs an extra monthly cost beyond the standard costs associated with sending email using Amazon SES. Each dedicated IP address incurs a separate monthly charge. For pricing information, see the Amazon SES pricing page.

**Control over Sender Reputation**

When you use dedicated IP addresses, your Amazon SES account is the only one that is able to send email from those addresses. For this reason, the sender reputation of the dedicated IP addresses that you lease is determined by your email sending practices.

**Ability to Isolate Sender Reputation**

By using dedicated IP addresses, you can isolate your sender reputation for different components of your email program. If you lease more than one dedicated IP address for use with Amazon SES, you can create dedicated IP pools—groups of dedicated IP addresses that can be used for sending specific types of email. For example, you can create one pool of dedicated IP addresses for sending marketing email, and another for sending transactional email. To learn more, see Creating Dedicated IP Pools (p. 162).

**Known, Unchanging IP Addresses**

When you use dedicated IP addresses, you can find the values of the addresses that send your mail in the Dedicated IPs page of the Amazon SES console. Dedicated IP addresses do not change.

With shared IP addresses, you do not know the IP addresses that Amazon SES uses to send your mail, and they can change at any time.

**Requesting and Relinquishing Dedicated IP Addresses**

This section describes how to request and relinquish dedicated IP addresses by submitting a request in the AWS Support Center. Note that you're charged an additional monthly fee for each dedicated IP address that you lease for use with Amazon SES. For more information about the costs associated with dedicated IP addresses, see Amazon SES Pricing.

**Best Practices for Working with Dedicated IP Addresses**

Although there's no minimum commitment, we recommend that you lease three or more dedicated IP addresses in order to maximize the delivery of your email.

Each AWS Region that Amazon SES is available in contains at least three Availability Zones (zones). When you lease more than one dedicated IP address, we distribute them evenly across the Availability Zones for your region.

If you create a dedicated IP pool (p. 162) that contains only one address, and the zone where that address is based is temporarily unavailable, emails that you attempt to send using that pool might not be sent. However, if you create a pool that contains three addresses, you can ensure that each address is in a separate zone. In this case, even in the unlikely event that two of the three zones are unavailable, you'll still be able to send email from the remaining address.

Service interruptions are rare, but it's a good practice to use redundant systems, especially if you send time-sensitive or mission-critical email. For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.
Requesting Dedicated IP Addresses

The following steps show how to request dedicated IP addresses by submitting an SES Sending Limits Increase case. You can use this process to request as many dedicated IP addresses as you need.

To request dedicated IP addresses

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. Open an SES Sending Limits Increase case in Support Center.
3. In the case submission form, make the following selections:
   - For **Regarding**, choose **Service Limit Increase**.
   - For **Limit Type**, choose **SES Sending Limits**.
   - For **Region**, choose the AWS Region in which you use Amazon SES.
     
     **Note**
     Dedicated IP addresses are unique to each AWS Region, so it is important to select the region in which you use Amazon SES.
     For example, if you have a dedicated IP address for use in the US West (Oregon) region, you will not be able to use that address to send email from the US East (N. Virginia) region.
     If you need to request dedicated IP addresses for use in more than one region, submit a separate request form for each region.
   - For **Limit**, choose **Desired Daily Sending Quota**.
   - For **New limit value**, enter the approximate number of emails you plan to send per day from the dedicated IP addresses that you are requesting.
   - For **Mail Type**, choose the type of email you plan to send using your dedicated IP address. If multiple values apply, choose the type that will make up the majority of your email sending.
   - For **Website URL**, type the URL of your website. Providing this information will help us better understand the type of content you plan to send.
   - For **My email sending complies with the AWS Service Terms and AUP**, choose the option that applies to your use case.
   - For **I only send to recipients who have specifically requested my mail**, choose the option that applies to your use case.
   - For **I have a process to handle bounces and complaints**, choose the option that applies to your use case.
   - For **Use Case Description**, state that you want to request dedicated IP addresses, and indicate the number of dedicated IP addresses that you are requesting. Next, describe the ways in which you will use dedicated IP addresses to send email using Amazon SES. Include information about why you want to use dedicated IP addresses as opposed to shared IP addresses; this information will help us better understand your use case.
   - For **Support Language**, choose your preferred language.
   - For **Contact method**, choose **Web**.

   When you finish, choose **Submit**.

Once you submit the form, we will evaluate your request. We will reply to your request in Support Center. If your request is granted, the reply will confirm that your dedicated IP addresses are now associated with your account.
Relinquish Dedicated IP Addresses

If you no longer need dedicated IP addresses that are assigned to your account, you can relinquish them by completing the following steps.

To relinquish dedicated IP addresses

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. Open an SES Sending Limits Increase case in Support Center.
3. In the case submission form, make the following selections:
   - For Regarding, choose Service Limit Increase.
   - For Limit Type, choose SES Sending Limits.
   - For Region, choose the AWS Region in which you use Amazon SES.
     Note
     Dedicated IP addresses are unique to each AWS Region, so it is important to select the region in which you use Amazon SES. If you need to relinquish dedicated IP addresses in more than one region, submit a separate request form for each region.
   - For Limit, choose Desired Daily Sending Quota.
   - For New limit value, enter any number. The number you enter here is not important—you will specify how many dedicated IPs you want to relinquish in the Use Case Description field.
   - For Use Case Description, state that you want to relinquish dedicated IP addresses, and then state the number of dedicated IP addresses you want to relinquish.
   - For Support Language, choose your preferred language.
   - For Contact method, choose Web.

When you finish, choose Submit.

After we evaluate your request, you will receive a reply within the case asking you to confirm that you want to release the number of dedicated IP addresses that you specified. You will receive a case reply confirming that your dedicated IP addresses have been released.

Warming up Dedicated IP Addresses

When determining whether to accept or reject a message, email service providers consider the reputation of the IP address that sent it. One of the factors that contributes to the reputation of an IP address is whether the address has a history of sending high-quality email. Email providers are less likely to accept mail from new IP addresses that have little or no history. Email sent from IP addresses with little or no history may end up in recipients' junk mail folders, or may be blocked altogether.

When you start sending email from a new IP address, you should gradually increase the amount of email you send from that address before using it to its full capacity. This process is called warming up the IP address.

The amount of time required to warm up an IP address varies between email providers. For some email providers, you can establish a positive reputation in around two weeks, while for others it may take up to six weeks. When warming up a new IP address, you should send emails to your most active users to ensure that your complaint rate remains low. You should also carefully examine your bounce messages and send less email if you receive a high number of blocking or throttling notifications. For information about monitoring your bounces, see Monitoring Your Amazon SES Sending Activity (p. 221).
Automatically Warm up Dedicated IP Addresses

When you request dedicated IP addresses, Amazon SES automatically warms them up to improve the delivery of emails you send. The automatic IP address warm-up feature is enabled by default.

The steps that happen during the automatic warm-up process depend on whether or not you already have dedicated IP addresses:

- When you request dedicated IP addresses for the first time, Amazon SES distributes your email sending between your dedicated IP addresses and a set of addresses that are shared with other Amazon SES customers. Amazon SES gradually increases the number of messages sent from your dedicated IP addresses over time.
- If you already have dedicated IP addresses, Amazon SES distributes your email sending between your existing dedicated IPs (which are already warmed up) and your new dedicated IPs (which are not warmed up). Amazon SES gradually increases the number of messages sent from your new dedicated IP addresses over time.

After you warm up a dedicated IP address, you should send around 1,000 emails every day to each email provider that you want to maintain a positive reputation with. You should perform this task on each dedicated IP address that you use with Amazon SES.

You should avoid sending large volumes of email immediately after the warm-up process is complete. Instead, slowly increase the number of emails you send until you reach your target volume. If an email provider sees a large, sudden increase in the number of emails being sent from an IP address, they may block or throttle the delivery of messages from that address.

Disable the Automatic Warm-up Process

When you purchase new dedicated IP addresses, Amazon SES automatically warms them up for you. If you prefer to warm up dedicated IP addresses yourself, you can disable the automatic warm-up feature.

**Important**

If you disable the automatic warm up feature, you are responsible for warming up your dedicated IP addresses yourself. If you send email from addresses that haven't been warmed up, you may experience poor delivery rates.

**To disable the automatic warm-up feature**

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation bar on the left, choose Dedicated IPs.
3. Clear the box next to Automatic IP warm-up.

Restart the Automatic Warm-up Process

You can restart the automatic IP warm-up process for a set of IP addresses that belong to a dedicated IP pool.

**To restart the automatic warm-up process**

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation bar on the left, choose Dedicated IPs.
3. In the dedicated IP pool for which you want to restart the warm-up process, choose Actions, and then choose Restart IP warm up.
The status of the automatic warm-up process is in the **Warm Up Status** column; when the warm-up process is finished, this column will say **Complete**.

## Creating Dedicated IP Pools

If you purchased several dedicated IP addresses to use with Amazon SES, you can create groups of those addresses. These groups are called **dedicated IP pools**. A common scenario is to create one pool of dedicated IP addresses for sending marketing communications, and another for sending transactional emails. Your sender reputation for transactional emails is then isolated from that of your marketing emails. In this scenario, if a marketing campaign generates a large number of complaints, the delivery of your transactional emails is not impacted.

This section contains procedures for creating dedicated IP pools.

**Note**

You can also create configuration sets that use a pool of IP addresses that are shared by all Amazon SES customers. The shared IP pool is useful in situations where you need to send email that doesn't align with your usual sending behaviors. For information about using the shared IP pool with a configuration set, see **Managing IP Pools** (p. 218).

### To create a dedicated IP pool using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at [https://console.aws.amazon.com/ses/](https://console.aws.amazon.com/ses/).
2. In the navigation pane on the left side of the screen, under **Email Sending**, choose **Dedicated IPs**.
3. Choose **Create a New IP Pool**.
4. On the **IP Pool Name** page, for **Pool name**, type a descriptive name for the dedicated IP pool, and then choose **Next**.
5. On the **Add Dedicated IPs** page, check the box next to each IP address you want to add to the pool, and then choose **Next**.

**Note**

Dedicated IP addresses that have not yet been assigned to a pool are listed under `ses-default-dedicated-pool`. A dedicated IP address can only belong to a single pool. If you select a dedicated IP address that is associated with a different pool, that setting is overwritten, and the address is instead associated with the pool you are creating.

6. On the **Assign to a configuration set** page, do one of the following:

   - Select **Add this pool to an existing configuration set** to associate the dedicated IP pool with an existing configuration set. Then, under **Existing configuration sets**, choose the configuration set with which you want to associate the IP pool.
   - Select **Create a new configuration set** to create a configuration set and associate the dedicated IP pool with it. For **Configuration set name**, type a descriptive name for the configuration set.

When you finish, choose **Next**.

7. On the **Review** page, verify the settings of the dedicated IP pool. When you are ready to create the IP pool, choose **Create**.

## Testing Amazon SES Email Sending

Amazon Simple Email Service (Amazon SES) provides a mailbox simulator that you can use to test how your application handles various email sending scenarios without affecting your sending quota or your
Testing Email Sending

bounce and complaint metrics. The Amazon SES mailbox simulator is a set of test email addresses. Each email address represents a specific scenario. You can send emails to the mailbox simulator when you want to:

- Test your application without having to create test "To" addresses.
- Test how your email sending program handles bounces, complaints, and out-of-the-office (OOTO) responses.
- See what happens when you email an address that is on the Amazon SES suppression list.
- Generate a bounce without putting a valid email address on the suppression list.
- Find your system's maximum throughput without using up your daily sending quota.
- Send test emails without affecting your email deliverability metrics for bounces and complaints.

To use the mailbox simulator, email the addresses and observe how your setup responds to the simulated scenarios. The following table lists each simulated scenario and the corresponding email address that you would use. The email addresses are not case-sensitive.

**Note**
You can only access the mailbox simulator by using Amazon SES. You cannot access it from an external mail server.

<table>
<thead>
<tr>
<th>Simulated scenario</th>
<th>Mailbox simulator email address</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Success</strong>—The recipient's ISP accepts your email. If you have set up delivery notifications as described in Monitoring Using Amazon SES Notifications (p. 226), Amazon SES sends you a delivery notification through Amazon Simple Notification Service (Amazon SNS). Otherwise, you will not receive any confirmation about this successful delivery other than the API return value.</td>
<td><a href="mailto:success@simulator.amazonses.com">success@simulator.amazonses.com</a></td>
</tr>
<tr>
<td><strong>Bounce</strong>—The recipient's ISP rejects your email with an SMTP 550 5.1.1 response code (&quot;Unknown User&quot;). Amazon SES generates a bounce notification and sends it to you via email or by using an Amazon SNS notification, depending on how you set up your system. This mailbox simulator email address will not be placed on the Amazon SES suppression list as one normally would when an email hard bounces. The bounce response that you receive from the mailbox simulator is compliant with RFC 3464. For information about how to receive bounce feedback, see Monitoring Using Amazon SES Notifications (p. 226).</td>
<td><a href="mailto:bounce@simulator.amazonses.com">bounce@simulator.amazonses.com</a></td>
</tr>
<tr>
<td><strong>Out of the Office</strong>—The recipient's ISP accepts your email and delivers it to the recipient's inbox. The ISP sends an out-of-the-office (OOTO) message to Amazon SES. Amazon SES then forwards the OOTO message to you via email or by using an Amazon SNS notification, depending on how you set up your system. The OOTO response that you receive from the Mailbox Simulator is compliant with RFC 3834. For information about how to set up your system to receive OOTO responses, follow the same instructions for setting up how Amazon SES sends you notifications in Monitoring Using Amazon SES Notifications (p. 226).</td>
<td><a href="mailto:ooto@simulator.amazonses.com">ooto@simulator.amazonses.com</a></td>
</tr>
<tr>
<td><strong>Complaint</strong>—The recipient's ISP accepts your email and delivers it to the recipient's inbox. The recipient, however,</td>
<td><a href="mailto:complaint@simulator.amazonses.com">complaint@simulator.amazonses.com</a></td>
</tr>
</tbody>
</table>
Simulated scenario

does not want to receive your message and clicks "Mark as Spam" within an email application that uses an ISP that sends a complaint response to Amazon SES. Amazon SES then forwards the complaint notification to you via email or by using an Amazon SNS notification, depending on how you set up your system. The complaint response that you receive from the mailbox simulator is compliant with RFC 5965. For information about how to receive bounce feedback, see Monitoring Using Amazon SES Notifications (p. 226).

Address on Suppression List—Amazon SES treats your email as a hard bounce because the address you are sending to is on the Amazon SES suppression list.

Mailbox simulator email address

suppressionlist@simulator.amazonses.com

Important

If you send an email to a mailbox simulator address other than the test addresses listed above, the unlisted address will be placed on the suppression list.

The mailbox simulator provides typical bounce, complaint, and OOTO responses. In the bounce scenario, multiple bounces from the same sending request are gathered into a single response. In practice, the response varies by ISP. To reduce your bounce and complaint rates, see the Amazon Simple Email Service Email Sending Best Practices whitepaper.

When you send emails to the mailbox simulator, you will be limited by your maximum send rate. You will also be billed for your emails. However, emails to the mailbox simulator will not affect your email deliverability metrics for bounces and complaints or count against your sending quota.

The mailbox simulator supports labeling, which enables you to send emails to the same mailbox simulator address in multiple ways, or to test your support for Variable Envelope Return Path (VERP). For example, you can send an email to bounce+label1@simulator.amazonses.com and bounce +label2@simulator.amazonses.com to test how your setup matches a bounce message with the undeliverable address that caused the bounce. For more information about VERP, see https://en.wikipedia.org/wiki/Variable_envelope_return_path.

You can send emails to the mailbox simulator even if you are in the sandbox.

Amazon SES and Security Protocols

This topic describes the security protocols that you can use when you connect to Amazon SES, as well as when Amazon SES delivers an email to a receiver.

Email Sender to Amazon SES

The security protocol that you use to connect to Amazon SES depends on whether you are using the Amazon SES API or the Amazon SES SMTP interface, as described next.

HTTP

If you are using the Amazon SES API (either directly or through an AWS SDK), then all communications are encrypted by TLS through the Amazon SES HTTPS endpoint. The Amazon SES HTTPS endpoint supports TLS 1.2, TLS 1.1, and TLS 1.0.
SMTP Interface

If you are accessing Amazon SES through the SMTP interface, you are required to encrypt your connection using Transport Layer Security (TLS). Note that TLS is often referred to by the name of its predecessor protocol, Secure Sockets Layer (SSL).

Amazon SES supports two mechanisms for establishing a TLS-encrypted connection: STARTTLS and TLS Wrapper.

- **STARTTLS**—STARTTLS is a means of upgrading an unencrypted connection to an encrypted connection. There are versions of STARTTLS for a variety of protocols; the SMTP version is defined in RFC 3207. For STARTTLS connections, Amazon SES supports TLS 1.2, TLS 1.1, TLS 1.0 and SSLv2Hello.

- **TLS Wrapper**—TLS Wrapper (also known as SMTPS or the Handshake Protocol) is a means of initiating an encrypted connection without first establishing an unencrypted connection. With TLS Wrapper, the Amazon SES SMTP endpoint does not perform TLS negotiation: it is the client's responsibility to connect to the endpoint using TLS, and to continue using TLS for the entire conversation. TLS Wrapper is an older protocol, but many clients still support it. For TLS Wrapper connections, Amazon SES supports TLS 1.2, TLS 1.1 and TLS 1.0.

For information about connecting to the Amazon SES SMTP interface using these methods, see Connecting to the Amazon SES SMTP Endpoint (p. 70).

Amazon SES to Receiver

Amazon SES sends messages over a TLS-protected connection (TLS version 1.0 only) by default. This method, called opportunistic TLS, means that when Amazon SES establishes an SMTP connection with a receiving mail server, Amazon SES upgrades the connection using the STARTTLS protocol if the receiving mail server supports TLS. If the receiving server does not advertise STARTTLS or if TLS negotiation fails, the connection proceeds in plaintext.

Amazon SES supports opportunistic TLS in all regions and you don't need to take any action to enable it.
Receiving Email with Amazon SES

Amazon Simple Email Service (Amazon SES) is a mail server that can both send and receive mail on your behalf. When you use Amazon SES to receive your mail, Amazon SES handles underlying mail-receiving operations, such as:

- communicating with other mail servers
- scanning for spam and viruses
- rejecting mail from untrusted sources
- accepting mail for recipients in your domain

When you receive email, Amazon SES processes it according to instructions you provide. For example, Amazon SES can deliver incoming mail to an Amazon S3 bucket, publish it to an Amazon SNS topic, or send it to Amazon WorkMail. You can also create rules that explicitly block or allow all messages from specific IP address ranges, or that automatically send bounce messages when messages are sent to specific email addresses.

The following sections contain the information you need to understand, set up, and use Amazon SES to receive your mail.

- Email-Receiving Concepts (p. 166)
- Getting Started Receiving Email (p. 168)
- Setting Up Email Receiving (p. 179)
- Managing Email Receiving (p. 198)

Amazon SES Email-Receiving Concepts

When you use Amazon SES as your email receiver, you must tell the service what to do with your mail. The primary method, which gives you fine-grained control over your mail, is to specify the actions to take based on the recipient. The other method is to block or allow mail based on the originating IP address. This topic describes both methods.

Recipient-Based Control

The primary way to control your incoming mail is to specify how mail is handled based on its recipient. For example, if you own example.com, you can specify that mail for user@example.com should bounce, and that all other mail for example.com and its subdomains should be delivered. The list of recipients you provide is called the condition.

You set up receipt rules to specify how to handle the mail when a condition is satisfied. A receipt rule consists of a condition and an ordered list of actions. If the recipient to whom the incoming mail is addressed matches a recipient specified in the condition, then Amazon SES performs the actions specified in the rule. The following actions are available:

- **S3 action**—Delivers the mail to an Amazon S3 bucket and, optionally, notifies you through Amazon SNS.
- **SNS action**—Publishes the mail to an Amazon SNS topic.

**Note**
The SNS action includes a complete copy of the email content in the Amazon SNS notifications. The other Amazon SNS notifications mentioned here simply notify you of email delivery; they contain information about the email, not the email content itself.
• **Lambda action**—Calls your code through a Lambda function and, optionally, notifies you through Amazon SNS.

• **Bounce action**—Rejects the email by returning a bounce response to the sender and, optionally, notifies you through Amazon SNS.

• **Stop action**—Terminates the evaluation of the receipt rule set and, optionally, notifies you through Amazon SNS.

• **Add header action**—Adds a header to the received email. You typically use this action only in combination with other actions.

• **WorkMail action**—Handles the mail with Amazon WorkMail. You will typically not use this action directly because Amazon WorkMail takes care of the setup.

Receipt rules are grouped together into receipt rule sets. You can define multiple receipt rule sets for your AWS account, but only one receipt rule set is active at any time. The following figure shows how receipt rules, receipt rule sets, and actions relate to each other.

**IP Address-Based Control**

You can control your mail flow on a broader level by setting up IP address filters. IP address filters are optional and enable you to specify whether to accept or reject mail originating from an IP address or range of IP addresses. Your IP address filters can include block lists (IP addresses from which you want to block incoming mail) and allow lists (IP addresses from which you want to always accept mail). IP address filters are useful for blocking spam. Amazon SES maintains its own block list of IP addresses known to send spam, but you can choose to receive mail from those IP addresses by adding them to your allow list.

**Note**

If you want to allow mail that originates from an Amazon EC2 IP address, you must add it to your allow list. All mail originating from Amazon EC2 is blocked by default.

**Email-Receiving Process**

When Amazon SES receives an email for your domain, the following events occur:

1. Amazon SES first looks at the IP address of the sender. Amazon SES allows the mail to pass this stage unless:
   - The IP address is in your block list.
   - The IP address is in the Amazon SES block list and not on your allow list.

2. Amazon SES examines your active receipt rule set to determine whether any of your receipt rules contain a condition that matches any of the incoming email’s recipients.

3. If there aren’t any matches, Amazon SES rejects the mail. Otherwise, Amazon SES accepts the mail.
4. If Amazon SES accepts the mail, it evaluates your active receipt rule set. All of the receipt rules that
match at least one of the recipient conditions are applied in the order that they are defined, unless an
action or a receipt rule explicitly terminates evaluation of the receipt rule set.

Now that you have an overview of the process, you can get started by going to Setting Up Email
Receiving (p. 179).

Getting Started Receiving Email with Amazon SES

In this tutorial, you will set up Amazon SES to receive email for a domain that you have registered with
Route 53, and deliver the email to an Amazon S3 bucket.

Topics
- Step 1: Before You Begin (p. 168)
- Step 2: Verify Your Domain (p. 169)
- Step 3: Set up a Receipt Rule (p. 172)
- Step 4: Send an Email (p. 176)
- Step 5: View the Received Email (p. 176)
- Step 6: Clean Up (p. 179)

Step 1: Before You Begin

Before you can start this tutorial, you must sign up for an AWS account (if you don't already have one),
and register the domain you want to receive email for with Route 53. The following sections describe
where to accomplish those tasks.

Sign Up

If you already have an AWS account, skip to the next section: Register a Domain with Route 53 (p. 168).
Otherwise, follow these steps.

To sign up for an AWS account

1. Open https://aws.amazon.com/, and then choose Create an AWS Account.

   Note
   This might be unavailable in your browser if you previously signed into the AWS
   Management Console. In that case, choose Sign in to a different account, and then choose
   Create a new AWS account.

2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone
   keypad.

Register a Domain with Route 53

In this tutorial, you will receive email for a domain that you have registered with Route 53. To get this
domain, you can either register a new domain or transfer an existing domain. Transferring an existing
domain can take several days.

Next step: Step 2: Verify Your Domain (p. 169)
Step 2: Verify Your Domain

Before you can receive email for a domain using Amazon SES, you must prove that you own the domain by verifying it with Amazon SES. Although Amazon SES enables you to verify single email addresses, you must verify a domain if you want to use Amazon SES for email receiving. You can verify and receive email with Amazon SES for any domain that you own, but it is easier to set up a domain that you have registered with Route 53.

To verify a domain with Amazon SES

1. Sign in to the AWS Management Console using the AWS account that you used to register your domain with Route 53, and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Identity Management, choose Domains. Then, in the content pane, choose Verify a New Domain.
3. Enter the name of the domain that is registered in Route 53, leave Generate DKIM Settings unselected (it is for email sending), and then choose Verify This Domain.
4. On the Verify a New Domain page, which displays the records you must add to your DNS server, choose Use Route 53.
Note
If you do not see **Use Route 53**, then your domain is not registered with Route 53, which is a prerequisite for this tutorial.

5. On the **Use Route 53** page, choose **Domain Verification Record**, choose **Email Receiving Record**, and choose the hosted zone you want to use.

**Important**
If you have already set up mail exchanger (MX) records for your domain, the next step will replace your old MX records with new records. MX records specify the mail server that you want to accept emails on behalf of your domain.
6. Choose **Create Record Sets**. You will go back to the **Domain Identities** list.

7. Wait a few minutes, and then refresh the **Domain Identities** list by using the refresh button near the top right of the content pane. Confirm that the status of the domain is **verified**.

Next step: **Step 3: Set up a Receipt Rule (p. 172)**
Step 3: Set up a Receipt Rule

To use Amazon SES as your email receiver, you must have at least one active receipt rule set. A receipt rule set is an ordered collection of receipt rules that specify what Amazon SES should do with mail it receives across all of your verified domains. Because you are setting up email receiving with Amazon SES for the first time, Amazon SES automatically creates a default receipt rule set for you, so you can jump into creating a receipt rule. The receipt rule you create will belong to the default receipt rule set.

To create a receipt rule

1. In the left navigation pane, under Email Receiving, choose Rule Sets. Then, in the content pane, choose Create a Receipt Rule.

2. On the Recipients page, choose Next Step. Because you aren't adding any recipients, this receipt rule will handle mail for all recipients in all of your verified domains.
3. From the Add action menu, choose S3.

4. For S3 bucket, choose Create S3 bucket.
5. For **Bucket Name**, enter a new bucket name. Bucket names must comply with the following requirements.

   - Can contain lowercase letters, numbers, periods (.), and hyphens (-).
   - Must be unique across all of AWS.
   - Must start with a number or letter.
   - Must be between 3 and 63 characters long.
   - Must not contain underscores (_), end with a hyphen, or be formatted as an IP address (e.g., 192.168.5.4).
   - Cannot contain two, adjacent periods or dashes next to periods.

Next, choose **Create Bucket**.

**Note**
Because you are creating the Amazon S3 bucket using the Amazon SES console, Amazon SES automatically sets up the policy required to give Amazon SES permission to write to the bucket. However, if you choose an Amazon S3 bucket that already exists, you must explicitly
give Amazon SES permission to write to the bucket by attaching a policy to the bucket using the Amazon S3 console or API.

6. Leave all other options at their default settings for the simplicity of this tutorial, and choose Next Step.

7. On the Rule Details page, for the rule name, type my-rule, leave all other options at their default settings, and then choose Next Step.

Next step: Step 4: Send an Email (p. 176)

Step 4: Send an Email

Sign in to the email client that you use to send email, and then send an email to an email address in the domain for which you set up email receiving.

Next step: Step 5: View the Received Email (p. 176)

Step 5: View the Received Email

1. In the upper left of the AWS Management Console, choose the AWS icon, and then, under Storage & Content Delivery, choose S3.
2. In the Amazon S3 console, choose the bucket that you created when you set up Amazon SES email receiving earlier in this tutorial.
3. In the bucket, find the received email. The name of the email will be an alphanumeric string. The bucket will also contain an item with the name AMAZON_SES_SETUP_NOTIFICATION, which you can ignore.

4. To download the email to your computer, choose the box to the left of the email, and then choose Download from the menu.
5. Open the email in a text editor. The email will be in a raw format, which is typically Multipurpose Internet Mail Extensions (MIME). To decode MIME, you must use your own application.

Next step: Step 6: Clean Up (p. 179)

**Step 6: Clean Up**

After you have completed this tutorial, you should clean up the following settings and resources to avoid incurring additional charges.

**Amazon SES Receipt Rule Set**

If you no longer want Amazon SES to receive mail for your domain, you must disable your active receipt rule set (p. 199).

**Amazon S3 Bucket**

If you no longer want the Amazon S3 bucket that you created, you can delete it. However, you cannot delete an Amazon S3 bucket that has items in it, so you must first delete the contents of the bucket, and then delete the bucket. For more information about deleting folders and buckets, see Delete an Object and Bucket in the *Amazon S3 Getting Started Guide*.

**Route 53 Domain**

If you no longer want Route 53 to register your domain, you can delete the registration or transfer your domain to another registrar.

**Setting Up Amazon SES Email Receiving**

This section describes what you need to do to configure Amazon SES to receive your mail. For example, you should first consider how you want to receive, filter, and process your mail, because those decisions
will affect how you configure Amazon SES. You also need to verify your domain with Amazon SES to prove that you own it, and point your domain to Amazon SES for incoming mail. Another step is to give Amazon SES permission to access any required AWS resources. Then you configure email receiving by creating a receipt rule set, receipt rules, and optionally, IP address filters.

These steps are explained in the following topics:
- Considering Your Use Case for Amazon SES Email Receiving (p. 180)
- Verifying Your Domain for Amazon SES Email Receiving (p. 182)
- Publishing an MX Record for Amazon SES Email Receiving (p. 182)
- Giving Permissions to Amazon SES for Email Receiving (p. 182)
- Creating IP Address Filters for Amazon SES Email Receiving (p. 184)
- Creating a Receipt Rule Set for Amazon SES Email Receiving (p. 185)
- Creating Receipt Rules for Amazon SES Email Receiving (p. 185)

To see where these tasks fit into the overall email-receiving process, see Email-Receiving Concepts (p. 166).

Considering Your Use Case for Amazon SES Email Receiving

Before you set up Amazon SES to receive your mail, you might find it helpful to consider the following questions.

Email Content

- **How do you want Amazon SES to pass you the email content?**
  
  Amazon SES can provide you the email content in two ways: it can store the emails in an Amazon S3 bucket that you specify, or it can send you an Amazon SNS notification that contains a copy of the email. Amazon SES delivers you the raw, unmodified email, which is typically in Multipurpose Internet Mail Extensions (MIME) format. For more information about MIME format, see RFC 2045.

- **How large of a limit on email size do you need?**
  
  If you choose to store emails in an Amazon S3 bucket, the maximum email size (including headers) is 30 MB. If you choose to receive your emails through Amazon SNS notifications, the maximum email size (including headers) is 150 KB.

- **How do you want to trigger the processing of your mail?**
  
  After your mail is delivered, you will want to process it with your own code. For example, your application might convert the base 64-encoded email into a displayable format and then make it available to an end user through an email client. There are a couple of ways you can start the process:
  - If your emails are delivered to Amazon S3, your application can listen for Amazon SNS notifications generated by S3 actions, extract the message ID of the email from the notifications, and then use the message ID to retrieve the email from Amazon S3.
  - Alternatively, you can incorporate email processing into your receipt rules by writing a Lambda function. In this case, your receipt rule should first write the email to Amazon S3, and then trigger the Lambda function. Lambda actions can be executed synchronously or asynchronously from within your receipt rules, depending on whether the Lambda function needs to return a result that influences how other actions are executed. We recommend that you use asynchronous execution unless synchronous is absolutely necessary for your use case. For more information about AWS Lambda, see the AWS Lambda Developer Guide.
• If your emails are delivered through an Amazon SNS notification by using the SNS action, your application can listen for Amazon SNS notifications, and then extract the email messages from the notifications.

• Do you want the emails to be encrypted?

Amazon SES integrates with AWS Key Management Service (AWS KMS) to optionally encrypt the mail it writes to your Amazon S3 bucket. Amazon SES uses client-side encryption to encrypt your mail before writing it to Amazon S3. This means that you must decrypt the content on your side after retrieving the mail from Amazon S3. The AWS SDK for Java and AWS SDK for Ruby provide a client that can handle the decryption for you. Amazon SES can encrypt the emails for you only if you choose for your emails to be delivered to an Amazon S3 bucket.

Unwanted Mail

• At what point in the email-receiving process do you want to reject unwanted mail?

You can reject emails at two points in the email-receiving process: during the SMTP conversation with the sender’s mail server, and after delivery when you can examine the email's properties. You are not billed for any mail that is rejected during the SMTP conversation, so it is to your advantage to reject as much unwanted mail as possible at that time. You can reject emails during the SMTP conversation with IP address filters and receipt rules, both of which are described in Email-Receiving Concepts (p. 166).

After the SMTP conversation, Amazon SES performs virus scanning, spam scanning, as well as authentication checks for DomainKeys Identified Mail (DKIM), Sender Policy Framework (SPF), and Domain-based Message Authentication, Reporting & Conformance (DMARC). Amazon SES makes the verdicts of all these tests available to you so you can decide if you trust the email. If you don’t trust the email, you can drop it or send a bounce response to the sender. You will be billed for the email because this decision point occurs after Amazon SES delivered the email to you. If you use Lambda to send a bounce response, you may also be billed for your usage of Lambda.

• Do you want to filter your emails based on any property other than recipient or IP address?

You can write complex message-matching conditions using Lambda functions (invoked as "RequestResponse") and then incorporate those functions into your receipt rules. The return value of the Lambda function determines whether the evaluation of the receipt rule and receipt rule set should continue or stop. For example, you can have a receipt rule that drops mail that Amazon SES flags as spam. You can also bounce messages that do not pass certain authentication tests, such as DKIM or DMARC. Code examples for both of these situations are available in Lambda Function Examples (p. 193).

Using Other AWS Services

• Have you set up the appropriate permissions?

If you want your mail to be delivered to an Amazon S3 bucket, published to an Amazon SNS topic you don’t own, trigger a Lambda function, or use a custom master AWS KMS key, you need to give Amazon SES permission to access those resources. To give Amazon SES access, you create policies on resources from the consoles or APIs for those AWS services. For more information Giving Permissions (p. 182).

Mail Streams

• How do you want to divide your mail stream?

Your domain most likely receives different classes of mail. For example, some of your domain's mail, such as an email to user@example.com, might be intended for a personal inbox. Other mail, such as an
email to unsubscribe@example.com, might be better directed to automated systems instead. You can use receipt rules to divide your incoming mail so that it can be processed differently. For information about how to set up receipt rules, see Creating Receipt Rules (p. 185).

Verifying Your Domain for Amazon SES Email Receiving

As with any domain you want to use for sending or receiving email with Amazon SES, you must first prove that you own it. The verification procedure, which includes initiating domain verification with Amazon SES and then publishing a TXT record to your DNS server, is described in Verifying Domains in Amazon SES (p. 52).

Note
Although Amazon SES enables you to verify single email addresses, you must verify a domain if you want to use Amazon SES for email receiving.

You can also start the domain verification process when you set up receipt rules in Creating Receipt Rules (p. 185). The recipient list will indicate which recipients are not verified, and enable you to initiate verification. In any case, you must complete domain verification by publishing a TXT record to your DNS server, as described in Amazon SES Domain Verification TXT Records (p. 56).

You can confirm that your email address or domain is verified by looking at its status in the Email Address Identities or Domain Identities list in the Amazon SES console or by using the Amazon SES GetIdentityVerificationAttributes API.

Publishing an MX Record for Amazon SES Email Receiving

A mail exchanger record (MX record) is a configuration that specifies which mail servers can accept email that is sent to your domain.

To have Amazon SES manage your incoming email, you add an MX record to your domain's DNS configuration. The MX record refers to the email receiving endpoint for the AWS Region where you use Amazon SES. For example, the endpoint for the US West (Oregon) region is inbound-smtp.us-west-2.amazonaws.com. See Email Receiving Endpoints (p. 409) for a complete list of endpoints.

The procedures for publishing an MX record depend on your DNS or hosting provider. See your provider's documentation for information about adding an MX record to the DNS configuration for your domain.

Giving Permissions to Amazon SES for Email Receiving

To enable Amazon SES to write emails to your Amazon S3 bucket, use an AWS KMS key to encrypt your emails, call your Lambda function, or publish to an Amazon SNS topic of another account, Amazon SES must have permission to access those resources. You give permission by attaching a policy to the resource. This topic provides example policies.

Give Amazon SES Permission to Write to Your Amazon S3 Bucket

When applied to an Amazon S3 bucket, the following policy gives Amazon SES permission to write to that bucket. For more information about creating receipt rules that transfer incoming email to Amazon S3, see S3 Action (p. 196). For more information about attaching policies to Amazon S3 buckets, see the Amazon Simple Storage Service Developer Guide.
Give Amazon SES Permission to Use Your AWS KMS Master Key

For Amazon SES to encrypt your emails, it must have permission to use the AWS KMS key that you specified when you set up your receipt rule. You can either use the default master key (aws/ses) in your account or a custom master key you create. If you use the default master key, you don't need to perform any steps to give Amazon SES permission to use it. If you use a custom master key, you need to give Amazon SES permission to use it by adding a statement to the key's policy. The policy statement includes conditions that are designed to ensure that Amazon SES can only use your custom master key when certain values are present in the request to AWS KMS; specifically:

- **aws:ses:source-account**—The AWS account ID on behalf of which Amazon SES received the email.
- **aws:ses:message-id**—The Amazon SES message ID of the received email.
- **aws:ses:rule-name**—The name of the receipt rule that was used to encrypt the email.

Paste the following policy statement into the key policy to permit Amazon SES to use your custom master key when Amazon SES receives email on behalf of your AWS account. Replace **AWSACCOUNTID** with your 12-digit AWS account ID.

```json
{
   "Sid": "AllowSESToEncryptMessagesBelongingToThisAccount",
   "Effect": "Allow",
   "Principal": { "Service": "ses.amazonaws.com" },
   "Action": [ "kms:Encrypt", "kms:GenerateDataKey*" ],
   "Resource": "*",
   "Condition": {
      "Null": { "kms:EncryptionContext:aws:ses:rule-name": "false",
                  "kms:EncryptionContext:aws:ses:message-id": "false" },
      "StringEquals": { "kms:EncryptionContext:aws:ses:source-account": "AWSACCOUNTID" }
   }
}
```

For more information about attaching policies to AWS KMS keys, see the **AWS Key Management Service Developer Guide**.
Give Amazon SES Permission to Invoke Your Lambda Function

To enable Amazon SES to call your Lambda function, you can either configure the Lambda function using the Amazon SES console during receipt-rule setup (in which case Amazon SES automatically adds the necessary permissions to the function) or you can use the AWS Lambda AddPermission API to attach a policy to the function. The following AddPermission API call gives Amazon SES permission to invoke your Lambda function. Replace AWSACCOUNTID with your 12-digit AWS account ID. For more information about attaching policies to Lambda functions, see the AWS Lambda Developer Guide.

```
{
    "Action": "lambda:InvokeFunction",
    "Principal": "ses.amazonaws.com",
    "SourceAccount": "AWSACCOUNTID",
    "StatementId": "GiveSESPermissionToInvokeFunction"
}
```

Give Amazon SES Permission to Publish to an Amazon SNS Topic of Another Account

If the Amazon SNS topic you want to use is owned by the same AWS account you are using for Amazon SES, no setup is required to allow Amazon SES to publish to the topic. However, if you want to publish notifications to a topic that you do not own, use the Amazon SNS console or API to attach a policy to the Amazon SNS topic. The following policy gives Amazon SES permission to publish to an Amazon SNS topic. Replace AWSACCOUNTID with your 12-digit AWS account ID, and TOPIC-NAME with the name of the Amazon SNS topic. For more information about writing policies for Amazon SNS topics, see the Amazon Simple Notification Service Developer Guide.

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": { "Service": "ses.amazonaws.com" },
            "Action": "SNS:Publish",
            "Resource": "arn:aws:sns:us-east-1:AWSACCOUNTID:TOPIC-NAME"
        }
    ]
}
```

Creating IP Address Filters for Amazon SES Email Receiving

An IP address filter enables you to optionally specify whether to accept or reject mail originating from an IP address or range of IP addresses.

You can use the Amazon SES console or the CreateReceiptFilter API to create an IP address filter.

**Note**

If you only want to receive mail from a finite list of known IP addresses, then set up a block list that contains 0.0.0.0/0, and set up an allow list that contains the IP addresses that you trust. This configuration blocks all IP addresses by default, and only allows mail from the IP addresses that you explicitly specify.
To create an IP address filter (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Email Receiving, choose IP Address Filters.
3. In the content pane, choose Create Filter.
4. For Filter Name, type a name for the IP address filter. The name must contain less than 64 alphanumeric, hyphen (-), underscore (_), and period (.) characters. The name must start and end with a letter or number.
5. For IP Address Range, type a single IP address or a range of IP addresses that you want to block or allow, specified in Classless Inter-Domain Routing (CIDR) notation. An example of a single IP address is 10.0.0.1. An example of a range of IP addresses is 10.0.0.1/24. For more information about CIDR notation, see RFC 2317.
6. For Policy Type, choose Allow or Block.
7. Choose Create Filter.

For information about how to use the CreateReceiptFilter API to create an IP address filter, see the Amazon Simple Email Service API Reference.

Creating a Receipt Rule Set for Amazon SES Email Receiving

A receipt rule set is an ordered collection of receipt rules that specify what Amazon SES should do with mail it receives across all of your domains. To use Amazon SES as your email receiver, you must create at least one receipt rule set for your account. You can set up multiple receipt rule sets, but only one receipt rule set in each AWS region can be active at any given time. For more information about the role of receipt rule sets in the email-receiving process, see Email-Receiving Concepts (p. 166).

Note
If you do not want to use Amazon SES as your email receiver, simply disable all of your receipt rule sets. For information about how to disable receipt rule sets, see Managing Receipt Rule Sets (p. 198).

You can use the Amazon SES console or API to create a receipt rule set.

• Using the Amazon SES console
  • Receipt rules exist in receipt rule sets only, so to create a receipt rule set, you can start by creating a receipt rule. For more information, see Creating Receipt Rules (p. 185). When you reach the end of this procedure, you can create a new receipt rule set.
  • Copy an existing receipt rule set as explained in Managing Receipt Rule Sets (p. 198).
  • In the left navigation pane, under Email Receiving, choose Rule Sets, and then choose Create a New Rule Set.

• Using the Amazon SES API—Use the CreateReceiptRuleSet API to create an empty receipt rule set, as described in the Amazon Simple Email Service API Reference. Then, you can use the Amazon SES console or the CreateReceiptRule API to add receipt rules to it.

Creating Receipt Rules for Amazon SES Email Receiving

Receipt rules let you specify what Amazon SES does with email it receives for the email addresses or domains you own. A receipt rule contains a condition and an ordered list of actions. If the recipient of
an incoming email matches a recipient specified in the conditions for the receipt rule, then Amazon SES performs the actions specified in that receipt rule. For more information about the role of receipt rules in the email-receiving process, see Email-Receiving Concepts (p. 166).

**Important**
To set up receipt rules, first verify a domain and publish an MX record on that domain. For more information about verifying domains, see Verifying Domains in Amazon SES (p. 52). For more information about publishing MX records, see the section called “Publishing an MX Record” (p. 182).

You can use the Amazon SES console or the CreateReceiptRule API operation to create receipt rules. This section provides procedures for creating a new receipt rule using the console. These procedures assume that your Amazon SES account does not contain any existing receipt rules.

### Setting Up a Receipt Rule

You can use the Amazon SES console or the CreateReceiptRule API to create rules.

**To create a receipt rule using the console**

1. Sign in to the AWS Management Console and open the Amazon SES console at [https://console.aws.amazon.com/ses/](https://console.aws.amazon.com/ses/).
2. In the left navigation pane, under Email Receiving, choose Rule Sets.
3. Choose Create a Receipt Rule.
4. Use the following procedure to add one or more recipients. Collectively, these recipients are the condition. You can have a maximum of 100 recipients per receipt rule.

   a. Under Recipients, specify the incoming email address or domain for which you want to set up a receipt rule. The following table uses the address user@example.com to show how to specify recipients.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Specify the following recipient...</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match a specific email address.</td>
<td><a href="mailto:user@example.com">user@example.com</a></td>
<td>Also matches variations of the address that contain labels (such as user <a href="mailto:+123@example.com">+123@example.com</a> and <a href="mailto:user+xyz@example.com">user+xyz@example.com</a>). However, if you specify an address that contains a label, only that specific address is matched.</td>
</tr>
<tr>
<td>Match all addresses within a domain, but not those within its subdomains.</td>
<td>example.com</td>
<td></td>
</tr>
<tr>
<td>Match all addresses within a specific subdomain, but not those within the parent domain.</td>
<td>subdomain.example.com</td>
<td></td>
</tr>
<tr>
<td>Match all addresses within all subdomains, but not those within the parent domain.</td>
<td>.example.com</td>
<td>Note the period (.) before the domain name.</td>
</tr>
</tbody>
</table>
If you want to...  | Specify the following recipient...  | Notes
---|---|---
Match all addresses within a domain, and all addresses within all of its subdomains. | example.com  
.example.com | Create two separate recipients: one with the domain name, and one with a period followed by the domain name.

| Match all recipients in all verified domains | [None] | Leave the recipient field blank.

**Important**
If multiple Amazon SES accounts receive email on a common domain (for example, if multiple teams in the same company each have separate Amazon SES accounts), Amazon SES processes all matching receipt rules simultaneously for each of those accounts. This behavior may result in a situation where one account generates a bounce, while another account accepts the email. We recommend that you coordinate with other teams in your organization that use Amazon SES to ensure that each account uses unique receipt rules, and that those rules do not overlap. In these situations, it is best to configure your receipt rules to use only email addresses or subdomains that are unique to your group or team.

b. Choose **Add Recipient**.

c. Repeat steps a and b for each recipient you want to add. When you finish adding recipients, choose **Next Step**.

5. Use the following procedure to add one or more actions to the receipt rule.

a. Choose an action from the menu.

b. Choose the action settings. For information about the options for each action, see **Action Options** (p. 188).

c. Add additional actions as needed, and then choose **Next Step**.

6. For **Rule Details**, use the following procedure to choose settings.

a. For **Rule Name**, type a name for the receipt rule. The name must contain less than 64 alphanumeric, hyphen (-), underscore (_), and period (.) characters. The name must start and end with a letter or number.

b. If you want to enable the receipt rule, leave the **Enabled** option selected.

c. If you want Amazon SES to reject any incoming emails that are not sent over a connection that is encrypted with Transport Layer Security (TLS), select **TLS**.

d. If you want Amazon SES to scan incoming emails for spam and viruses, select **Enable Spam and Virus Scanning**.

7. For **Rule Set**, choose an existing receipt rule set or click **Create New Rule Set**.

8. For **Rule Position**, choose where to place the receipt rule in the ordered list of receipt rules. The receipt rules are evaluated sequentially.

9. Choose **Next Step**, and then choose **Create Rule**.

For information about how to use the CreateReceiptRule API to create rules, see the **Amazon Simple Email Service API Reference**.
Action Options

Each receipt rule for Amazon SES email receiving contains an ordered list of actions. The overall setup procedure for receipt rules is described in Creating Receipt Rules for Amazon SES Email Receiving (p. 185). This section describes the specific options for each action type.

The action types are the following:

- Add Header Action (p. 188)
- Bounce Action (p. 188)
- Lambda Action (p. 189)
- S3 Action (p. 196)
- SNS Action (p. 197)
- Stop Action (p. 197)
- WorkMail Action (p. 197)

Add Header Action

The Add Header action adds a custom header to the received email. You typically use this action only in combination with another action. This action has the following options.

- Header name—The name of the header to add. It must be between 1 and 50 characters, inclusive, and consist of alphanumeric (a-z, A-Z, 0-9) characters and dashes only.
- Header value—The value of the header to add. It must be less than 2048 characters, and must not contain newline characters ("\r" or "\n").

Bounce Action

The Bounce action rejects the email by returning a bounce response to the sender and, optionally, notifies you through Amazon SNS. This action has the following options.

- SMTP Reply Code—The SMTP reply code, as defined by RFC 5321.
- SMTP Status Code—The SMTP enhanced status code, as defined by RFC 3463.
- Message—Human-readable text to include in the bounce email.
- Reply Sender—The email address of the sender of the bounced email. This is the address from which the bounce email will be sent. It must be verified with Amazon SES.
- SNS Topic—The name or ARN of the Amazon SNS topic to optionally notify when a bounce email is sent. An example of an Amazon SNS topic ARN is arn:aws:sns:us-west-2:123456789012:MyTopic. You can also create an Amazon SNS topic when you set up your action by choosing Create SNS Topic. For more information about Amazon SNS topics, see the Amazon Simple Notification Service Developer Guide.

Note
The Amazon SNS topic you choose must be in the same AWS region as the Amazon SES endpoint you use to receive email.

You can type in your own values for these fields, or you can choose a template that fills in the SMTP Reply Code, SMTP Status Code, and Message fields with values based on the bounce reason. The following templates are available:

- Mailbox Does Not Exist— SMTP Reply Code = 550, SMTP Status Code = 5.1.1
- Message Too Large— SMTP Reply Code = 552, SMTP Status Code = 5.3.4
• **Message Full**—SMTP Reply Code = 552, SMTP Status Code = 5.2.2
• **Message Content Rejected**—SMTP Reply Code = 500, SMTP Status Code = 5.6.1
• **Unknown Failure**—SMTP Reply Code = 554, SMTP Status Code = 5.0.0
• **Temporary Failure**—SMTP Reply Code = 450, SMTP Status Code = 4.0.0

For additional bounce codes that you might use by typing custom values in the fields, see RFC 3463.

**Lambda Action**

The Lambda action calls your code through a Lambda function and, optionally, notifies you through Amazon SNS. This action has the following options.

• **Lambda function**—The ARN of the Lambda function. An example of a Lambda function ARN is `arn:aws:lambda:us-west-2:account-id:function:MyFunction`. For information about AWS Lambda, see the AWS Lambda Developer Guide.

• **Invocation type**—The invocation type of the Lambda function. An invocation type of `RequestResponse` means that the execution of the function will immediately result in a response, and a value of `Event` means that the function will be invoked asynchronously. We recommend that you use `Event` invocation type unless synchronous execution is absolutely necessary for your use case.

  **Note**
  There is a 30-second timeout on `RequestResponse` invocations.

  For information about AWS Lambda invocation types, see the AWS Lambda Developer Guide.

• **SNS Topic**—The name or ARN of the Amazon SNS topic to notify when the specified Lambda function is triggered. An example of an Amazon SNS topic ARN is `arn:aws:sns:us-west-2:123456789012:MyTopic`. You can also create an Amazon SNS topic when you set up your action by choosing Create SNS Topic. For more information about Amazon SNS topics, see the Amazon Simple Notification Service Developer Guide.

  **Note**
  The Amazon SNS topic you choose must be in the same AWS region as the Amazon SES endpoint you use to receive email.

**Writing Your Lambda Function**

To process your email, your Lambda function can be invoked asynchronously (that is, using the `Event` invocation type). The event object passed to your Lambda function will contain metadata pertaining to the inbound email event. You can also use the metadata to access the message content from your Amazon S3 bucket.

If you want to actually control the mail flow, your Lambda function must be invoked synchronously (that is, using the `RequestResponse` invocation type) and your Lambda function must call the `callback` method with two arguments: the first argument is `null`, and the second argument is a `disposition` property that is set to either `STOP_RULE`, `STOP_RULE_SET`, or `CONTINUE`. If the second argument is `null` or does not have a valid `disposition` property, the mail flow continues and further actions and rules are processed, which is the same as with `CONTINUE`.

For example, you can stop the receipt rule set by writing the following line at the end of your Lambda function code:

```java
callback( null, { "disposition" : "STOP_RULE_SET" });
```

For AWS Lambda code samples, see Lambda Function Examples (p. 193). For examples of high-level use cases, see Use Case Examples (p. 190).
Input Format

Amazon SES passes information to the Lambda function in JSON format. The top-level object contains a Records array, which is populated with properties eventSource, eventVersion, and ses. The ses object contains receipt and mail objects, which are in exactly the same format as in the Amazon SNS notifications described in Notification Contents (p. 205).

The following is a high-level view of the structure of the input that Amazon SES provides to the Lambda function.

```
{
  "Records": [
    {
      "eventSource": "aws:ses",
      "eventVersion": "1.0",
      "ses": {
        "receipt": {
          "<same contents as SNS notification>
        },
        "mail": {
          "<same contents as SNS notification>
        }
      }
    }
  ]
}
```

Return Values

Your Lambda function can control mail flow by returning one of the following values:

- **STOP_RULE**—No further actions in the current receipt rule will be processed, but further receipt rules can be processed.
- **STOP_RULE_SET**—No further actions or receipt rules will be processed.
- **CONTINUE** or any other invalid value—This means that further actions and receipt rules can be processed.

Use Case Examples

The following examples outline some rules that you might set up to use Lambda function outcomes to control your mail flow. For demonstration purposes, many of these examples use the S3 action as the outcome.

**Use Case 1: Drops Spam Across All Domains**

This example demonstrates a global rule that drops spam across all of your domains. Rules 2 and 3 are included to show that you can apply domain-specific rules after the spam is dropped over all the domains.

**Rule 1**

Recipient list: Empty. This rule will therefore apply to all recipients under all of your verified domains.

Actions

1. Lambda action (synchronous) that returns STOP_RULE_SET if the email is spam. Otherwise, it returns CONTINUE. See the example Lambda function for dropping spam in Lambda Function Examples (p. 193).
Rule 2

Recipient list: example1.com

Actions

1. Any action.

Rule 3

Recipient list: example2.com

Actions

1. Any action.

Use Case 2: Bounces Spam Across All Domains

This example demonstrates a global rule that bounces spam across all of your domains. Rules 2 and 3 are included to show that you can apply domain-specific rules after the spam is bounced over all the domains.

Rule 1

Recipient list: Empty. This rule will therefore apply to all recipients under all of your verified domains.

Actions

1. Lambda action (synchronous) that returns CONTINUE if the email is spam. Otherwise, it returns STOP_RULE.
2. Bounce action ("500 5.6.1. Message content rejected").
3. Stop action.

Rule 2

Recipient list: example1.com

Actions

1. Any action

Rule 3

Recipient list: example2.com

Actions

1. Any action

Use Case 3: Applies the Most Specific Rule

This example demonstrates how you can use the Stop action to prevent emails from being processed by multiple rules. In this example, you have one rule for a specific address, and another rule for all email addresses under the domain. By using the Stop action, messages that match the rule for the specific email address are not processed by the more generic rule that applies to the domain.
Rule 1

Recipient list: user@example.com

Actions
1. Lambda action (asynchronous).
2. Stop action.

Rule 2

Recipient list: example.com

Actions
1. Any action.

Use Case 4: Logs Mail Events to CloudWatch

This example demonstrates how to keep an audit log of all mail going through your system before saving the mail to Amazon SES.

Rule 1

Recipient list: example.com

Actions
1. Lambda action (asynchronous) that writes the event object to a CloudWatch log. The example Lambda functions in Lambda Function Examples (p. 193) log to CloudWatch.
2. S3 action.

Use Case 5: Drops Mail That Fails DKIM

This example demonstrates how you can save all incoming email to an Amazon S3 bucket, but only send email that goes to a specific email address, and passes DKIM, to your automated email application.

Rule 1

Recipient list: example.com

Actions
1. S3 action.
2. Lambda action (synchronous) that returns STOP_RULE_SET if the message fails DKIM. Otherwise, it returns CONTINUE.

Rule 2

Recipient list: support@example.com

Actions
1. Lambda action (asynchronous) that triggers the automated application.
Use Case 6: Filters Mail Based on Subject Line

This example demonstrates how you can drop all of a domain's incoming mail that contains the word "discount" in the subject line, and then process mail intended for an automated system one way, and process mail addressed to all other recipients in the domain a different way.

Rule 1

Recipient list: example.com

Actions

1. Lambda action (synchronous) that returns STOP_RULE_SET if the subject line contains the word "discount". Otherwise, it returns CONTINUE.

Rule 2

Recipient list: support@example.com

Actions

1. S3 action with bucket 1.
2. Lambda action (asynchronous) that triggers the automated application.
3. Stop action.

Rule 3

Recipient list: example.com

Actions

1. S3 action with bucket 2.
2. Lambda action (asynchronous) that processes email for the rest of the domain.

Lambda Function Examples

This topic contains examples of Lambda functions that control mail flow.

Example 1: Drops Spam

This example stops processing messages that have at least one spam indicator.

```javascript
exports.handler = function(event, context, callback) {
    console.log('Spam filter');

    var sesNotification = event.Records[0].ses;
    console.log("SES Notification:\n", JSON.stringify(sesNotification, null, 2));

    // Check if any spam check failed
    if (sesNotification.receipt.spfVerdict.status === 'FAIL'
        || sesNotification.receipt.dkimVerdict.status === 'FAIL'
        || sesNotification.receipt.spamVerdict.status === 'FAIL'
        || sesNotification.receipt.virusVerdict.status === 'FAIL') {
        console.log('Dropping spam');
        // Stop processing rule set, dropping message
        callback(null, {'disposition':'STOP_RULE_SET'});
    }
};
```
Example 2: Continues if Particular Header

This example continues processing the current rule only if the email contains a specific header value.

```javascript
exports.handler = function(event, context, callback) {
  console.log('Header matcher');
  var sesNotification = event.Records[0].ses;
  console.log("SES Notification:\n", JSON.stringify(sesNotification, null, 2));

  // Iterate over the headers
  for (var index in sesNotification.mail.headers) {
    var header = sesNotification.mail.headers[index];
    // Examine the header values
    if (header.name === 'X-Header' && header.value === 'X-Value') {
      console.log('Found header with value.');
      callback(null, null);
      return;
    }
  }

  // Stop processing the rule if the header value wasn't found
  callback(null, {'disposition':'STOP_RULE'});
};
```

Example 3: Retrieves Email from Amazon S3

This example gets the raw email from Amazon S3 and processes it.

**Note**
You must first write the email to Amazon S3 using an S3 Action.

```javascript
var AWS = require('aws-sdk');
var s3 = new AWS.S3();
var bucketName = '<YOUR BUCKET GOES HERE>Appending';
exports.handler = function(event, context, callback) {
  console.log('Process email');
  var sesNotification = event.Records[0].ses;
  console.log("SES Notification:\n", JSON.stringify(sesNotification, null, 2));

  // Retrieve the email from your bucket
  s3.getObject({
    Bucket: bucketName,
    Key: sesNotification.mail.messageId
  }, function(err, data) {
    if (err) {
      console.log(err, err.stack);
      callback(err);
    } else {
      console.log("Raw email:\n" + data.Body);
      // Custom email processing goes here
      callback(null, null);
  }
};
```
Example 4: Bounces Messages that Fail DMARC Authentication

This example sends a bounce message if an incoming email fails DMARC authentication.

**Note**
When using this example, set the value of the `emailDomain` environment variable to your email receiving domain.

```javascript
'use strict';
const AWS = require('aws-sdk');

// Assign the emailDomain environment variable to a constant.
const emailDomain = process.env.emailDomain;

exports.handler = (event, context, callback) => {
  console.log('Spam filter starting');
  const sesNotification = event.Records[0].ses;
  const messageId = sesNotification.mail.messageId;
  const receipt = sesNotification.receipt;

  console.log('Processing message:', messageId);
  if (receipt.dmarcVerdict.status === 'FAIL'
      && receipt.dmarcPolicy.status === 'REJECT') {
    // The values that make up the body of the bounce message.
    const sendBounceParams = {
      BounceSender: `mailer-daemon@${emailDomain}`,
      OriginalMessageId: messageId,
      MessageDsn: {
        ReportingMta: `dns; ${emailDomain}`,
        ArrivalDate: new Date(),
        ExtensionFields: [],
      },
      // Include custom text explaining why the email was bounced.
      Explanation: "Unauthenticated email is not accepted due to the sending domain's DMARC policy.",
      BouncedRecipientInfoList: receipt.recipients.map((recipient) => ({
        Recipient: recipient,
        // Bounce with 550 5.6.1 Message content rejected
        BounceType: 'ContentRejected',
      })),
    },

    console.log('Bouncing message with parameters:');
    console.log(JSON.stringify(sendBounceParams, null, 2));
  // Try to send the bounce.
    new AWS.SES().sendBounce(sendBounceParams, (err, data) => {
      // If something goes wrong, log the issue.
      if (err) {
        console.log(`An error occurred while sending bounce for message: ${messageId}`, err);
        callback(err);
      // Otherwise, log the message ID for the bounce email.
      } else {
        console.log('Bounce for message ${messageId} sent, bounce message ID: ${data.MessageId}');
      }
    });
  }
};
```
// Stop processing additional receipt rules in the rule set.
callback(null, {
    disposition: 'stop_rule_set',
  });

// If the DMARC verdict is anything else (PASS, QUARANTINE or GRAY), accept
// the message and process remaining receipt rules in the rule set.

S3 Action

The S3 action delivers the mail to an Amazon S3 bucket and, optionally, notifies you through Amazon SNS. This action has the following options.

• **S3 Bucket**—The name of the Amazon S3 bucket to which to save received emails. You can also create a new Amazon S3 bucket when you set up your action by choosing Create S3 Bucket. Amazon SES provides you the raw, unmodified email, which is typically in Multipurpose Internet Mail Extensions (MIME) format. For more information about MIME format, see RFC 2045.

  **Important**
  When you save your emails to an Amazon S3 bucket, the maximum email size (including headers) is 30 MB.

• **Object Key Prefix**—A key name prefix to use within the Amazon S3 bucket. Key name prefixes enable you to organize your Amazon S3 bucket in a folder structure. For example, if you use Email as your Object Key Prefix, your emails will appear in your Amazon S3 bucket in a folder named Email.

• **KMS Key (if "Encrypt Message" is selected in the Amazon SES console)**—The customer master key that Amazon SES should use to encrypt your emails before saving them to the Amazon S3 bucket. You can use the default master key or a custom master key you created in AWS KMS.

  **Note**
  The master key you choose must be in the same AWS region as the Amazon SES endpoint you use to receive email.

  • To use the default master key, choose aws/ses when you set up the receipt rule in the Amazon SES console. If you use the Amazon SES API, you can specify the default master key by providing an ARN in the form of arn:aws:kms:REGION:AWSACCOUNTID:alias/aws/ses. For example, if your AWS account ID is 123456789012 and you want to use the default master key in the US West (Oregon) region, the ARN of the default master key would be arn:aws:kms:us-west-2:123456789012:alias/aws/ses. If you use the default master key, you don't need to perform any extra steps to give Amazon SES permission to use the key.

  • To use a custom master key you created in AWS KMS, provide the ARN of the master key and ensure that you add a statement to your key's policy to give Amazon SES permission to use it. For more information about giving permissions, see Giving Permissions to Amazon SES for Email Receiving (p. 182).

For more information about using AWS KMS with Amazon SES, see the AWS Key Management Service Developer Guide. If you do not specify a master key in the console or API, Amazon SES will not encrypt your emails.

  **Important**
  Your mail is encrypted by Amazon SES using the Amazon S3 encryption client before the mail is submitted to Amazon S3 for storage. It is not encrypted using Amazon S3 server-side encryption. This means that you must use the Amazon S3 encryption client to decrypt the email after retrieving it from Amazon S3, as the service has no access to use your AWS KMS keys for decryption. This encryption client is available in the AWS SDK for Java and the AWS
SDK for Ruby. For more information about client-side encryption using AWS KMS master keys, see the Amazon Simple Storage Service Developer Guide.

• **SNS Topic**—The name or ARN of the Amazon SNS topic to notify when an email is saved to the Amazon S3 bucket. An example of an Amazon SNS topic ARN is `arn:aws:sns:us-west-2:123456789012:MyTopic`. You can also create an Amazon SNS topic when you set up your action by choosing **Create SNS Topic**. For more information about Amazon SNS topics, see the Amazon Simple Notification Service Developer Guide.

  **Note**
  The Amazon SNS topic you choose must be in the same AWS region as the Amazon SES endpoint you use to receive email.

**SNS Action**

The **SNS** action publishes the mail using an Amazon SNS notification. The notification includes the complete email content. This action has the following options.

• **SNS Topic**—The name or ARN of the Amazon SNS topic to which to publish the emails. The Amazon SNS notifications will contain a raw, unmodified copy of the email, which is typically in Multipurpose Internet Mail Extensions (MIME) format. For more information about MIME format, see RFC 2045.

  **Important**
  If you choose to receive your emails through Amazon SNS notifications, the maximum email size (including headers) is 150 KB. Larger emails will bounce. If you anticipate emails larger than this size, save the emails to an Amazon S3 bucket instead.

  An example of an Amazon SNS topic ARN is `arn:aws:sns:us-west-2:123456789012:MyTopic`. You can also create an Amazon SNS topic when you set up your action by choosing **Create SNS Topic**. For more information about Amazon SNS topics, see the Amazon Simple Notification Service Developer Guide.

  **Note**
  The Amazon SNS topic you choose must be in the same AWS region as the Amazon SES endpoint you use to receive email.

• **Encoding**—The encoding to use for the email within the Amazon SNS notification. UTF-8 is easier to use, but may not preserve all special characters when a message was encoded with a different encoding format. Base64 preserves all special characters. For information about UTF-8 and Base64, see RFC 3629 and RFC 4648, respectively.

**Stop Action**

The **Stop** action terminates the evaluation of the receipt rule set and, optionally, notifies you through Amazon SNS. This action has the following options.

• **SNS Topic**—The name or ARN of the Amazon SNS topic to notify when the Stop action is performed. An example of an Amazon SNS topic ARN is `arn:aws:sns:us-west-2:123456789012:MyTopic`. You can also create an Amazon SNS topic when you set up your action by choosing **Create SNS Topic**. For more information about Amazon SNS topics, see the Amazon Simple Notification Service Developer Guide.

  **Note**
  The Amazon SNS topic you choose must be in the same AWS region as the Amazon SES endpoint you use to receive email.

**WorkMail Action**

The **WorkMail** action integrates with Amazon WorkMail. If Amazon WorkMail performs all of your email processing, you will typically not use this action directly because Amazon WorkMail takes care of the setup. This action has the following options.
• **Organization ARN**—The ARN of the Amazon WorkMail organization. Amazon WorkMail organization ARNs are in the form `arn:aws:workmail:region:account_ID:organization/organization_ID`, where:
  - `region` is the region in which you are using Amazon SES and Amazon WorkMail. (You must use them from the same region.) An example is `us-west-2`.
  - `account_ID` is the AWS account ID. You can find your AWS account ID on the Account page of the AWS Management Console.
  - `organization_ID` is a unique identifier that Amazon WorkMail generates when you create an organization. You can find the organization ID in the Amazon WorkMail console on the Organization Settings page of your organization.

An example of a complete Amazon WorkMail organization ARN is `arn:aws:workmail:us-west-2:123456789012:organization/m-68755160c4cb4e29a2b2f8fb58f359d7`. For information about Amazon WorkMail organizations, see the Amazon WorkMail Administrator Guide.

• **SNS Topic**—The name or ARN of the Amazon SNS topic to notify when the Amazon WorkMail action is taken. An example of an Amazon SNS topic ARN is `arn:aws:sns:us-west-2:123456789012:MyTopic`. You can also create an Amazon SNS topic when you set up your action by choosing Create SNS Topic. For more information about Amazon SNS topics, see the Amazon Simple Notification Service Developer Guide.

  **Note**
  The Amazon SNS topic you choose must be in the same AWS region as the Amazon SES endpoint you use to receive email.

---

Managing Amazon SES Email Receiving

After you create your receipt rule sets, receipt rules, and IP address filters, you can use the Amazon SES console or API to edit, delete, and perform other operations. You can also examine the Amazon SNS notifications you receive, and use Amazon CloudWatch to view your error metrics.

**Topics in this section:**
- Managing Receipt Rule Sets for Amazon SES Email Receiving (p. 198)
- Managing Receipt Rules for Amazon SES Email Receiving (p. 201)
- Managing IP Address Filters for Amazon SES Email Receiving (p. 203)
- Viewing Metrics for Amazon SES Email Receiving (p. 204)
- Using Notifications for Amazon SES Email Receiving (p. 204)

Managing Receipt Rule Sets for Amazon SES Email Receiving

After you create a receipt rule set as described in Creating a Receipt Rule Set (p. 185), you can update it as needed. Although editing a receipt rule set usually consists of editing individual receipt rules as described in Managing Receipt Rules (p. 201), you can also delete, activate, disable, and copy receipt rule sets. Additionally, you can reorder the receipt rules in a receipt rule set. These operations are described in the following sections.

**Topics in this section:**
- Deleting a Receipt Rule Set (p. 199)
- Activating and Disabling a Receipt Rule Set (p. 199)
- Copying a Receipt Rule Set (p. 200)
Deleting a Receipt Rule Set

You can use the Amazon SES console or the `DeleteReceiptRuleSet` API to delete a receipt rule set.

**Note**
You cannot delete the receipt rule set that is currently active.

**To delete a receipt rule set (console)**

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Email Receiving, choose Rule Sets.
3. In the Inactive Rule Sets list, select the receipt rule set that you want to delete.
4. From the Actions menu, choose Delete, and then confirm that you want to delete the receipt rule set.

For information about how to use the `DeleteReceiptRuleSet` API to delete a receipt rule set, see the Amazon Simple Email Service API Reference.

Activating and Disabling a Receipt Rule Set

Each receipt rule set is in one of two states: active or disabled. Only one of your receipt rule sets can be active at any given time. Disabled receipt rule sets can be useful in cases where you want to make changes to your active receipt rule set, but you do not want those changes to be active until you are sure your updates are correct. In that case, you can copy the active receipt rule set and make changes to the copied, disabled receipt rule set. After you're satisfied with the changes, you can activate the copied receipt rule set. When you activate a receipt rule set, all other receipt rule sets are disabled automatically.

**Note**
To disable email receiving through Amazon SES completely, disable all of your receipt rule sets.

You can use the Amazon SES console or the `SetActiveReceiptRuleSet` API to control which rule set is active.

**To activate a disabled receipt rule set (console)**

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Email Receiving, choose Rule Sets.
3. In the Inactive Rule Sets list, select the receipt rule set that you want to activate.
4. Choose Set as Active Rule Set.

**To disable the active receipt rule set (console)**

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Email Receiving, choose Rule Sets.
3. Under Active Rule Set, choose Disable Active Rule Set, and then confirm that you want to disable the receipt rule set.

For information about how to use the `SetActiveReceiptRuleSet` API to activate or disable a rule set, see the Amazon Simple Email Service API Reference.
Copying a Receipt Rule Set

You can use the Amazon SES console or the CloneReceiptRuleSet API to copy a receipt rule set. If you use the Amazon SES console, the procedure differs slightly, depending on whether the receipt rule set you want to copy is active or disabled.

To copy the active receipt rule set (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Email Receiving, choose Rule Sets.
3. In the content pane, choose Copy Active Rule Set.
4. In the Copy Rule Set dialog box, type the name you want to assign to the copied receipt rule set.
5. Choose Copy Rule Set. The copied receipt rule set will appear in the Inactive Rule Sets list.

To copy a disabled receipt rule set (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Email Receiving, choose Rule Sets.
3. In the Inactive Rule Sets list, select the receipt rule set that you want to copy.
4. From the Actions menu, choose Copy.
5. In the Copy Rule Set dialog box, type the name you want to assign to the copied receipt rule set.

For information about how to use the CloneReceiptRuleSet API to copy a receipt rule set, see the Amazon Simple Email Service API Reference.

Reordering Receipt Rules

You can use the Amazon SES console or the ReorderReceiptRuleSet API to reorder receipt rules in a receipt rule set. If you use the Amazon SES console, the procedure differs slightly, depending on whether the receipt rule set is active or disabled.

To reorder receipt rules in the active receipt rule set (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Email Receiving, choose Rule Sets.
3. In the content pane, choose View Active Rule Set.
5. Use the up and down arrows next to the receipt rule names to reorder the receipt rules, and then choose Save Order.

To reorder receipt rules in a disabled receipt rule set (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Email Receiving, choose Rule Sets.
3. In the Inactive Rule Sets list, select the receipt rule set.
5. Use the up and down arrows next to the receipt rule names to reorder the receipt rules, and then choose **Save Order**.

For information about how to use the ReorderReceiptRuleSet API to reorder receipt rules in a receipt rule set, see the Amazon Simple Email Service API Reference.

**Managing Receipt Rules for Amazon SES Email Receiving**

In addition to creating receipt rules as described in Creating Receipt Rules (p. 185), you can edit, delete, enable, disable, copy, and set the position of a receipt rule in its receipt rule set, as described in the following sections.

Note

The instructions in this section assume that the receipt rule is in the active receipt rule set. To edit the receipt rules of a disabled receipt rule set, choose a receipt rule set from the **Inactive Rule Sets** list. From there, the instructions for editing receipt rules are the same as for the active receipt rule set.

Topics in this section:

- Editing a Receipt Rule (p. 201)
- Deleting a Receipt Rule (p. 201)
- Enabling and Disabling a Receipt Rule (p. 202)
- Copying a Receipt Rule (p. 202)
- Setting the Position of a Receipt Rule (p. 203)

**Editing a Receipt Rule**

You can use the Amazon SES console or the Amazon SES API to edit a receipt rule. It is easier to use the Amazon SES console.

To edit a receipt rule (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under **Email Receiving**, choose **Rule Sets**.
3. In the content pane, choose **View Active Rule Set** or choose a receipt rule set from the **Inactive Rule Sets** list.
4. In the details pane, choose the receipt rule you want to edit.
5. In the **Edit Rule** pane, edit the policy, and then choose **Save Rule**.

If you want to use the Amazon SES API instead, use the DescribeReceiptRule API to retrieve the rule, use a text editor to edit the rule, and then use the UpdateReceiptRule API to overwrite the previous version of the rule. For more information, see the Amazon Simple Email Service API Reference.

**Deleting a Receipt Rule**

You can use the Amazon SES console or the DeleteReceiptRule API to delete a receipt rule.

To delete a receipt rule (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Email Receiving, choose Rule Sets.
3. In the content pane, choose View Active Rule Set or choose a receipt rule set from the Inactive Rule Sets list.
4. In the details pane, select the receipt rule.
5. From the Actions menu, choose Delete, and then confirm that you want to delete the receipt rule.

For information about how to use the DeleteReceiptRule API to delete a rule, see the Amazon Simple Email Service API Reference.

Enabling and Disabling a Receipt Rule

You can use the Amazon SES console or the Amazon SES API to enable or disable a receipt rule. It is easier to use the Amazon SES console.

To enable or disable a receipt rule (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Email Receiving, choose Rule Sets.
3. In the content pane, choose View Active Rule Set or choose a receipt rule set from the Inactive Rule Sets list.
4. In the details pane, choose the receipt rule you want to edit.
5. In the Edit Rule pane, select or clear Enabled, and then choose Save Rule.

If you want to use the Amazon SES API instead, you can use the DescribeReceiptRule API to retrieve the receipt rule, use a text editor to edit the receipt rule's Enabled field, and then use the UpdateReceiptRule API to overwrite the previous version of the receipt rule. For more information, see the Amazon Simple Email Service API Reference.

Copying a Receipt Rule

You can use the Amazon SES console or the Amazon SES API to copy a receipt rule. It is easier to use the Amazon SES console.

To copy a receipt rule (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Email Receiving, choose Rule Sets.
3. In the content pane, choose View Active Rule Set or choose a receipt rule set from the Inactive Rule Sets list.
4. In the details pane, select the receipt rule.
5. From the Actions menu, choose Copy Rule.
6. In the Copy Rule dialog box, type a new receipt rule name and select the destination receipt rule set. The new receipt rule will be inserted at the beginning of the receipt rule set, and it will initially be disabled.

If you want to use the Amazon SES API instead, you can use the DescribeReceiptRule API to retrieve the receipt rule, use a text editor to edit the receipt rule's name and receipt rule set (if desired), and then pass that receipt rule to the CreateReceiptRule API. For more information, see the Amazon Simple Email Service API Reference.
Setting the Position of a Receipt Rule

You can use the Amazon SES console or the SetReceiptRulePosition API to change the position of a receipt rule in the receipt rule set.

To set the position of a receipt rule (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Email Receiving, choose Rule Sets.
3. In the content pane, choose View Active Rule Set or choose a receipt rule set from the Inactive Rule Sets list.
4. In the content pane, choose Reorder Rules.
5. Use the up and down arrows next to the receipt rule names to reorder the receipt rules, and then choose Save Order.

For information about how to use the SetReceiptRulePosition API to change the position of a receipt rule in the receipt rule set, see the Amazon Simple Email Service API Reference.

Managing IP Address Filters for Amazon SES Email Receiving

In addition to creating IP address filters as explained in Creating IP Address Filters (p. 184), you can view and delete them, as described in the following sections.

Viewing IP Address Filters

You can use the Amazon SES console or the ListReceiptFilters API to get a list of your IP address filters.

To view your IP address filters (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Email Receiving, choose IP Address Filters. You will see a list of your IP address filters.

For information about how to use the ListReceiptFilters API to get a list of your IP address filters, see the Amazon Simple Email Service API Reference.

Deleting an IP Address Filter

You can use the Amazon SES console or the DeleteReceiptFilter API to delete an IP address filter.

To delete an IP address filter (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, under Email Receiving, choose IP Address Filters.
3. In the details pane, select the IP address filter.
4. Choose Delete, and then confirm that you want to delete the IP address filter.
For information about how to use the `DeleteReceiptFilter` API to delete an IP address filter, see the Amazon Simple Email Service API Reference.

### Viewing Metrics for Amazon SES Email Receiving

You can use Amazon CloudWatch (CloudWatch) to view failure metrics for your receipt rules. You'll find the metrics under **SES/Rule Metrics**.

There are two failure metrics:

- **PublishFailure** – Amazon SES encountered an error when it tried to execute the actions you configured.
- **PublishExpired** – Amazon SES encountered an error when it tried to execute the actions you configured, and Amazon SES will no longer retry to deliver the email. This failure can be permanent or transient. Amazon SES will no longer retry because the action did not succeed within four hours.

These errors can occur, for example, if you deleted or revoked permissions to an Amazon S3 bucket, Amazon SNS topic, or Lambda function that an action in one of your receipt rules was configured to use.

**Important**

Changes you make to fix your receipt rule set will apply only to emails that Amazon SES receives after the update. Emails are always evaluated against the receipt rule set that was in place at the time the email was received.

The following figure shows the metrics in the CloudWatch console.

![CloudWatch Metrics](image)

### Using Notifications for Amazon SES Email Receiving

When you receive an email, Amazon SES executes the rules in the active receipt rule set. You can configure receipt rules to send you notifications using Amazon SNS. Your receipt rules can send two different types of notifications:
• **Notifications sent from SNS actions** – When you add an SNS (p. 197) action to a receipt rule, it sends information about the email. If the message is 150KB or smaller, this notification type also includes the complete MIME body of the email.

• **Notifications sent from other action types** – When you add any other action type (including Bounce (p. 188), Lambda (p. 189), Stop Rule Set (p. 197), or WorkMail (p. 197) actions) to a receipt rule, you can optionally specify an Amazon SNS topic. If you do, you will receive notifications when these actions are performed. These notifications contain information about the email, but do not contain the content of the email.

This section describes the contents of these notifications, and provides an example of each type of notification.

**Topics in this section:**
- Contents of Notifications for Amazon SES Email Receiving (p. 205)
- Examples of Notifications for Amazon SES Email Receiving (p. 210)

### Contents of Notifications for Amazon SES Email Receiving

All notifications for email receiving are published to Amazon Simple Notification Service (Amazon SNS) topics in JavaScript Object Notation (JSON) format.

**Top-Level JSON Object**

The top-level JSON object contains the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>notificationType</td>
<td>String that specifies the notification type. This value will always be Received.</td>
</tr>
<tr>
<td>receipt</td>
<td>Object that contains information about the email delivery.</td>
</tr>
<tr>
<td>mail</td>
<td>Object that contains information about the email to which the notification pertains.</td>
</tr>
<tr>
<td>content</td>
<td>String that contains the raw, unmodified email, which is typically in Multipurpose Internet Mail Extensions (MIME) format. For more information about MIME format, see RFC 2045.</td>
</tr>
</tbody>
</table>

**Note**

This field is present only if the notification was triggered by an SNS action. Notifications triggered by all other actions do not contain this field.

**receipt Object**

The receipt object has the following fields.
### Field Name | Description
--- | ---
**action** | Object that encapsulates information about the action that was executed. For a list of possible values, see action Object (p. 206).
**dkimVerdict** | Object that indicates whether the DomainKeys Identified Mail (DKIM) check passed. For a list of possible values, see dkimVerdict Object (p. 207).
**dmarcPolicy** | Object that indicates the Domain-based Message Authentication, Reporting & Conformance (DMARC) settings for the sending domain. This field only appears if the value of dmarcVerdict is FAIL. For a list of possible values, see dmarcPolicy Object (p. 208).
**dmarcVerdict** | Object that indicates whether the Domain-based Message Authentication, Reporting & Conformance (DMARC) check passed. For a list of possible values, see dmarcVerdict Object (p. 208).
**processingTimeMillis** | String that specifies the period, in milliseconds, from the time Amazon SES received the message to the time it triggered the action.
**recipients** | A list of recipients that were matched by the active receipt rule. The addresses listed here may differ from those listed by the destination field in the section called “mail Object” (p. 210).
**spamVerdict** | Object that indicates whether the message is spam. For a list of possible values, see spamVerdict Object (p. 209).
**spfVerdict** | Object that indicates whether the Sender Policy Framework (SPF) check passed. For a list of possible values, see spfVerdict Object (p. 209).
**timestamp** | String that specifies the date and time at which the action was triggered, in ISO 8601 format.
**virusVerdict** | Object that indicates whether the message contains a virus. For a list of possible values, see virusVerdict Object (p. 209).

### action Object

The action object has the following fields.

| Field Name | Description |
--- | ---|
**type** | String that indicates the type of action that was executed. Possible values are S3, SNS, Bounce, Lambda, Stop, and WorkMail.
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>topicArn</td>
<td>String that contains the Amazon Resource Name (ARN) of the Amazon SNS topic to which the notification was published.</td>
</tr>
<tr>
<td>bucketName</td>
<td>String that contains the name of the Amazon S3 bucket to which the message was published. Present only for the S3 action type.</td>
</tr>
<tr>
<td>objectKey</td>
<td>String that contains a name that uniquely identifies the email in the Amazon S3 bucket. This is the same as the messageId in the section called “mail Object” (p. 210). Present only for the S3 action type.</td>
</tr>
<tr>
<td>smtpReplyCode</td>
<td>String that contains the SMTP reply code, as defined by RFC 5321. Present only for the bounce action type.</td>
</tr>
<tr>
<td>statusCode</td>
<td>String that contains the SMTP enhanced status code, as defined by RFC 3463. Present only for the bounce action type.</td>
</tr>
<tr>
<td>message</td>
<td>String that contains the human-readable text to include in the bounce message. Present only for the bounce action type.</td>
</tr>
<tr>
<td>sender</td>
<td>String that contains the email address of the sender of the email that bounced. This is the address from which the bounce message was sent. Present only for the bounce action type.</td>
</tr>
<tr>
<td>functionArn</td>
<td>String that contains the ARN of the Lambda function that was triggered. Present only for the Lambda action type.</td>
</tr>
<tr>
<td>invocationType</td>
<td>String that contains the invocation type of the Lambda function. Possible values are RequestResponse and Event. Present only for the Lambda action type.</td>
</tr>
<tr>
<td>organizationArn</td>
<td>String that contains the ARN of the Amazon WorkMail organization. Present only for the WorkMail action type.</td>
</tr>
</tbody>
</table>

**dkimVerdict Object**

The `dkimVerdict` object has the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>String that contains the DKIM verdict. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• PASS – The message passed DKIM authentication.</td>
</tr>
</tbody>
</table>
Using Notifications

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAIL</td>
<td>The message failed DKIM authentication.</td>
</tr>
<tr>
<td>GRAY</td>
<td>The message is not DKIM-signed.</td>
</tr>
<tr>
<td>PROCESSING_FAILED</td>
<td>There is an issue that prevents Amazon SES from checking the DKIM signature. For example, DNS queries are failing or the DKIM signature header is not formatted properly.</td>
</tr>
</tbody>
</table>

### dmarcPolicy Object

The `dmarcPolicy` object has the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>String that contains additional information about sending domains for emails that fail DMARC authentication. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• NONE – The owner of the sending domain requests that no specific action be taken on messages that fail DMARC authentication.</td>
</tr>
<tr>
<td></td>
<td>• QUARANTINE – The owner of the sending domain requests that messages that fail DMARC authentication be treated by receivers as suspicious.</td>
</tr>
<tr>
<td></td>
<td>• REJECT – The owner of the sending domain requests that messages that fail DMARC authentication be rejected.</td>
</tr>
</tbody>
</table>

### dmarcVerdict Object

The `dmarcVerdict` object has the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>String that contains the DMARC verdict. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• PASS – The message passed DMARC authentication.</td>
</tr>
<tr>
<td></td>
<td>• FAIL – The message failed DMARC authentication.</td>
</tr>
<tr>
<td></td>
<td>• GRAY – The message failed DMARC authentication, and the sending domain does not have a DMARC policy, or uses the p=none policy.</td>
</tr>
<tr>
<td></td>
<td>• PROCESSING_FAILED – There is an issue that prevents Amazon SES from providing a DMARC verdict.</td>
</tr>
</tbody>
</table>
spamVerdict Object

The spamVerdict object has the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>String that contains the result of spam scanning. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• PASS – The spam scan determined that the message is unlikely to contain spam.</td>
</tr>
<tr>
<td></td>
<td>• FAIL – The spam scan determined that the message is likely to contain spam.</td>
</tr>
<tr>
<td></td>
<td>• GRAY – Amazon SES scanned the email but could not determine with confidence whether it is spam.</td>
</tr>
<tr>
<td></td>
<td>• PROCESSING_FAILED – Amazon SES was unable to scan the email. For example, the email is not a valid MIME message.</td>
</tr>
</tbody>
</table>

spfVerdict Object

The spfVerdict object has the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>String that contains the SPF verdict. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• PASS – The message passed SPF authentication.</td>
</tr>
<tr>
<td></td>
<td>• FAIL – The message failed SPF authentication.</td>
</tr>
<tr>
<td></td>
<td>• GRAY – There is no SPF policy under the domain used in the MAIL FROM command.</td>
</tr>
<tr>
<td></td>
<td>• PROCESSING_FAILED – There is an issue that prevents Amazon SES from checking the SPF record. For example, DNS queries are failing.</td>
</tr>
</tbody>
</table>

virusVerdict Object

The virusVerdict object has the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>String that contains the result of virus scanning. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• PASS – The message does not contain a virus.</td>
</tr>
<tr>
<td></td>
<td>• FAIL – The message contains a virus.</td>
</tr>
<tr>
<td></td>
<td>• GRAY – Amazon SES scanned the email but could not determine with confidence whether it contains a virus.</td>
</tr>
</tbody>
</table>
### Field Name | Description
--- | ---
PROCESSING_FAILED | Amazon SES is unable to scan the content of the email. For example, the email is not a valid MIME message.

### mail Object

The `mail` object has the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>destination</td>
<td>A complete list of all recipient addresses (including To: and CC: recipients) from the MIME headers of the incoming email.</td>
</tr>
<tr>
<td>messageId</td>
<td>String that contains the unique ID assigned to the email by Amazon SES. If the email was delivered to Amazon S3, the message ID is also the Amazon S3 object key that was used to write the message to your Amazon S3 bucket.</td>
</tr>
<tr>
<td>source</td>
<td>String that contains the email address from which the email was sent (the envelope MAIL FROM address).</td>
</tr>
<tr>
<td>timestamp</td>
<td>String that contains the time at which the email was received, in ISO8601 format.</td>
</tr>
<tr>
<td>headers</td>
<td>A list of Amazon SES headers and your custom headers. Each header in the list has a name field and a value field.</td>
</tr>
<tr>
<td>commonHeaders</td>
<td>A list of headers common to all emails. Each header in the list is composed of a name and a value.</td>
</tr>
<tr>
<td>headersTruncated</td>
<td>String that specifies whether the headers were truncated in the notification, which will happen if the headers are larger than 10 KB. Possible values are <code>true</code> and <code>false</code>.</td>
</tr>
</tbody>
</table>

### Examples of Notifications for Amazon SES Email Receiving

This section includes examples of the following types of notifications:

- A notification sent as a result of an SNS action. (p. 210)
- A notification sent as a result of another type of action (p. 213) (an alert notification).

### Notification of an SNS action

This section contains an example of an SNS action notification. Unlike the alert notification shown previously, it includes a content section that contains the email, which is typically in Multipurpose Internet Mail Extensions (MIME) format.
{
    "notificationType": "Received",
    "receipt": {
        "timestamp": "2015-09-11T20:32:33.936Z",
        "processingTimeMillis": 222,
        "recipients": [
            "recipient@example.com"
        ],
        "spamsVerdict": {
            "status": "PASS"
        },
        "virusVerdict": {
            "status": "PASS"
        },
        "spfVerdict": {
            "status": "PASS"
        },
        "dkimVerdict": {
            "status": "PASS"
        },
        "action": {
            "type": "SNS",
            "topicArn": "arn:aws:sns:us-east-1:012345678912:example-topic"
        }
    },
    "mail": {
        "timestamp": "2015-09-11T20:32:33.936Z",
        "source": "61967230-7A45-4A9D-BEC9-87CBCF2211C9@example.com",
        "messageId": "d6iitobk75ur44p8kdnnp7g2n800",
        "destination": [
            "recipient@example.com"
        ],
        "headersTruncated": false,
        "headers": [
            {
                "name": "Return-Path",
                "value": "<0000014fbe1c09cf-7cb9f704-7531-4e53-89a1-5fa9744f5eb6-000000@amazonses.com>"
            },
            {
                "name": "Received",
                "value": "from a9-183.smtp-out.amazonses.com (a9-183.smtp-out.amazonses.com [54.240.9.183]) by inbound-smtp.us-east-1.amazonaws.com with SMTP id d6iitobk75ur44p8kdnnp7g2n800 for recipient@example.com; Fri, 11 Sep 2015 20:32:33 +0000 (UTC)"
            },
            {
                "name": "DKIM-Signature",
                "value": "v=1; a=rsa-sha256; q=dns/txt; c=relaxed/simple;
                s=amazonses.com; t=1442003552;
                b=04ukUDSFqghqibv+eP0DHI1k6j0J2akrzuopr6LBe6sUuvqpBkig56Uw29FbJF
                hlx30v7DeYVmlc38stqwsF8ivcajXp0sXRC1cW9z8x875J041rC1Aj7V7EgBmqdpVpFX
                4hHst1XPyXwmgdHImUuh8oZEKpVqir6bHHzz+zg="
            },
            {
                "name": "From",
                "value": "sender@example.com"
            },
            {
                "name": "To",
                "value": "recipient@example.com"
            }
        ]
    }
}
Example content
Alert Notification

This section contains an example of an Amazon SNS notification that can be triggered by an S3 action. Notifications triggered by Lambda actions, bounce actions, stop actions, and WorkMail actions are similar. Although the notification contains information about the email, it does not contain the content of the email itself.

```json
{
    "notificationType": "Received",
    "receipt": {
        "timestamp": "2015-09-11T20:32:33.936Z",
        "processingTimeMillis": 406,
        "recipients": [
            "recipient@example.com"
        ],
        "spamVerdict": {
            "status": "PASS"
        },
        "virusVerdict": {
            "status": "PASS"
        },
        "spfVerdict": {
            "status": "PASS"
        },
        "dkimVerdict": {
            "status": "PASS"
        },
        "action": {
            "type": "S3",
            "topicArn": "arn:aws:sns:us-east-1:012345678912:example-topic",
            "bucketName": "my-S3-bucket",
            "objectKey": "\email"
        }
    },
    "mail": {
        "timestamp": "2015-09-11T20:32:33.936Z",
        "source": "0000014fbe1c09cf-7cb9f704-7531-4e53-89a1-5fa9744f5eb6-00000000amazonses.com",
        "messageId": "d6iitobk75ur44p8kdnnp7g2n800",
        "destination": ["recipient@example.com"],
        "headersTruncated": false,
        "headers": [
            {
                "name": "Return-Path",
                "value": "<0000014fbe1c09cf-7cb9f704-7531-4e53-89a1-5fa9744f5eb6-00000000amazonses.com>"
            },
            {
                "name": "Received",
                "value": "from a9-183.smtp-out.amazonses.com (a9-183.smtp-out.amazonses.com [54.240.9.183]) by inbound-smtp.us-east-1.amazonaws.com with SMTP id d6iitobk75ur44p8kdnnp7g2n800 for recipient@example.com; Fri, 11 Sep 2015 20:32:33 +0000 (UTC)"
            },
            {
                "name": "DKIM-Signature",
                "value": "v=1; a=rsa-sha256; q=dns/txt; c=relaxed/simple; s=amazonses; d=amazonses.com; t=1442003552; b=q7ntbf4gcmlwjq322ax3p6g6z6t7dfe4kga54zmwq2k1o6z89k7f6m4k8ni6x3c6mzy5ou3kqz1kzcw8alwz22d7hrp4z4sbgm26t36ram9a3juz5tkn6x9y5h9c0znx1y6wvq8k"
            }
        ]
    }
}
```
{ "name": "From", "value": "sender@example.com" }
{ "name": "To", "value": "recipient@example.com" }
{ "name": "Subject", "value": "Example subject" }
{ "name": "MIME-Version", "value": "1.0" }
{ "name": "Content-Type", "value": "text/plain; charset=UTF-8" }
{ "name": "Content-Transfer-Encoding", "value": "7bit" }
{ "name": "Date", "value": "Fri, 11 Sep 2015 20:32:32 +0000" }
{ "name": "Message-ID", "value": "<61967230-7A45-4A9D-BEC9-87C6CF2211C9@example.com>" }
{ "name": "X-SES-Outgoing", "value": "2015.09.11-54.240.9.183" }
{ "name": "Feedback-ID", "value": "1.us-east-1.Krv2FKpFdwV+KUYw3Qd6wcpPJ4Sv/pOOpEpShn2u2o4=:AmazonSES" }
}
"commonHeaders": { "returnPath": "0000014fbe1c09cf-7cb9f704-7531-4e53-89a1-5fa9744f5eb6-000000@amazonses.com", "from": [ "sender@example.com" ], "date": "Fri, 11 Sep 2015 20:32:32 +0000", "to": [ "recipient@example.com" ], "messageId": "<61967230-7A45-4A9D-BEC9-87C6CF2211C9@example.com>" }, "subject": "Example subject" }
Using Amazon SES Configuration Sets

Configuration sets are groups of rules that you can apply to the emails you send using Amazon SES. You apply a configuration set to an email by including a reference to the configuration set in the headers of the email. When you apply a configuration set to an email, all of the rules in that configuration set are applied to the email. For more information about specifying configuration sets in your emails, see Specifying a Configuration Set When You Send Email (p. 219).

You can use configuration sets to apply the following types of rules to your emails:

- **Event publishing** – Amazon SES can track the number of send, delivery, open, click, bounce, and complaint events for each email you send. You can use event publishing to send information about these events to other AWS services. For example, you can send your email metrics to an Amazon Kinesis Data Firehose destination, and then analyze it using Amazon Kinesis Data Analytics. Alternatively, you can send bounce and complaint information to Amazon SNS and receive notifications immediately when those events occur.

- **IP pool management** – If you lease dedicated IP addresses to use with Amazon SES, you can create groups of these addresses, called *dedicated IP pools*. You can then associate these dedicated IP pools with configuration sets. A common use case is to create one pool of dedicated IP addresses for sending marketing communications, and another for sending transactional emails. Your sender reputation for transactional emails is then isolated from that of your marketing emails.

Configuration sets can contain one, both, or neither of these types of rules.

To learn more about managing configuration sets and their related components, see the following topics:

- Managing Amazon SES Configuration Sets (p. 215)
- Managing Amazon SES Event Destinations (p. 216)
- Managing IP Pools (p. 218)

Managing Amazon SES Configuration Sets

This section contains procedures for viewing a list of your existing configuration sets, viewing the details of those configuration sets, and deleting configuration sets.

**Viewing a List of Your Configuration Sets**

You can use the Amazon SES console or you can use the `ListConfigurationSets` API to view a list of your configuration sets.

**To view your configuration sets using the Amazon SES console**

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose **Configuration Sets**.
In the details pane, you will see a list of your configuration sets.

For information about how to use the ListConfigurationSets API to list your configuration sets, see the Amazon Simple Email Service API Reference.

**Viewing the Details of a Configuration Set**

You can use the Amazon SES console to view the details of a configuration set, or you can use the DescribeConfigurationSet API to describe a configuration set.

**Viewing the details of a configuration set using the Amazon SES console**

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose Configuration Sets.
3. In the details pane, choose the expand icon next to the configuration set.
   
   You will see the details of the configuration set.

For information about how to use the DescribeConfigurationSet API to describe a configuration set, see the Amazon Simple Email Service API Reference.

**Deleting a Configuration Set**

You can use the Amazon SES console or the DeleteConfigurationSet API to delete a configuration set.

**To delete a configuration set using the Amazon SES console**

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose Configuration Sets.
3. In the details pane, choose the configuration set.
4. From the Actions menu, choose Delete, and then confirm that you want to delete the configuration set.

For information about how to use the DeleteConfigurationSet API to delete a configuration set, see the Amazon Simple Email Service API Reference.

**Managing Amazon SES Event Destinations**

Event destinations allow you to publish email sending metrics—including the numbers of sends, deliveries, opens, clicks, bounces, and complaints—to other AWS products. To learn more about setting up event publishing, see the section called “Monitoring Using Event Publishing” (p. 247).

**Updating an Event Destination**

You can use the Amazon SES console or the UpdateConfigurationSetEventDestination API to update an event destination.
To update an event destination (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose Configuration Sets.
3. In the configuration set list, choose the configuration set that contains the event destination that you want to update.
4. In the Destination list, to the right of the destination you want to edit, choose the edit icon (pencil).
5. Edit the event destination details, and then choose Save.
6. To exit the Edit Configuration Set page, use the back button of your browser.

For information about how to use the UpdateConfigurationSetEventDestination API to update an event destination, see the Amazon Simple Email Service API Reference.

Deleting an Event Destination

You can use the Amazon SES console or the DeleteConfigurationSetEventDestination API to delete an event destination.

To delete an event destination (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose Configuration Sets.
3. In the configuration set list, choose the configuration set that contains the event destination that you want to delete.
4. In the Destination list, choose the delete icon (trash bin).
5. Confirm that you want to delete the configuration set.
6. To exit the Edit Configuration Set page, use the back button of your browser.

For information about how to use the DeleteConfigurationSetEventDestination API to delete an event destination, see the Amazon Simple Email Service API Reference.

Enabling or Disabling an Event Destination

You can use the Amazon SES console or the UpdateConfigurationSetEventDestination API to enable or disable an event destination.

To enable or disable an event destination (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose Configuration Sets.
3. In the configuration set list, choose the configuration set that contains the event destination that you want to enable or disable.
4. In the Destination list, to the right of the destination you want to edit, choose the edit icon (pencil).
5. Select or deselect Enabled, and then choose Save.
Managing IP Pools

You can use IP pools to create groups of dedicated IP addresses for sending specific types of email. You can also use a pool of IP addresses that are shared by all Amazon SES customers.

Assigning an IP Pool to an Existing Configuration Set

You can use the Amazon SES console to associate an IP pool with an existing configuration set.

To assign an IP pool to a configuration set

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose Configuration Sets.
3. In the list of configuration sets, choose the configuration set that you want to associate with an IP pool.
4. On the Sending IP pool tab, for Pool name, choose from one of the following options:
   - A specific dedicated IP pool – When you select an existing dedicated IP pool, emails that use the configuration set are sent using only the dedicated IP addresses that belong to that pool. For procedures for creating new IP pools, see Creating Dedicated IP Pools (p. 162).
   - ses-default-dedicated-pool – This pool contains all of the dedicated IP addresses for your account that do not already belong to an IP pool. If you send an email using a configuration set that is not associated with a pool, that email is sent from one of the addresses in the default pool.
   - ses-shared-pool – This pool contains a large set of IP addresses that are shared among all Amazon SES customers. This option may be useful when you need to send email that doesn't align with your usual sending behaviors.
5. When you are finished, choose Assign.

Modifying IP Pool Assignments

You can also use the Amazon SES console to assign a different pool to a configuration set that is already associated with a pool. Assigning a different pool to a configuration set overwrites the previous association.

To edit an IP pool assignment

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose Configuration Sets.
3. In the list of configuration sets, choose the configuration set that you want to modify.
4. On the Sending IP pool tab, under Assign an IP pool, choose the edit icon (✏).
5. For Pool name, select the pool that you want to use, and then choose Assign.
Specifying a Configuration Set When You Send Email

To use a configuration set when sending an email, you must pass the name of the configuration set in the headers of the email. All of the Amazon SES email sending methods—including the AWS CLI, the AWS SDKs, and the Amazon SES SMTP interface (p. 66)—allow you to pass a configuration set in the headers of the email you send.

If you are using the SMTP interface (p. 66) or the SendRawEmail API operation, you can specify a configuration set by including the following header in your email (replacing ConfigSet with the name of the configuration set you want to use):

```
X-SES-CONFIGURATION-SET: ConfigSet
```

This guide includes code examples for sending email using Postfix, the AWS SDKs, and the Amazon SES SMTP interface. Each of these examples includes a method of specifying a configuration set. To see step-by-step procedures for sending emails that include references to configuration sets, see the following:

- Integrating Amazon SES with Postfix (p. 82)
- Send an Email Through Amazon SES Using an AWS SDK (p. 28)
- Send an Email Through Amazon SES Using SMTP (p. 19)

You can find additional code examples in the Amazon SES Code Examples (p. 379) section.

Exporting Reputation Metrics for a Configuration Set to CloudWatch

Amazon SES automatically exports information about the overall bounce and complaint rates for your entire account to Amazon CloudWatch. You can use these metrics to create alarms in CloudWatch, or to automatically pause email sending using a Lambda function.

You can also export reputation metrics for individual configuration sets to CloudWatch. Exporting reputation data at the configuration set level gives you more control over your sender reputation.

This section includes procedures for exporting reputation data for individual configuration sets to CloudWatch by using the Amazon SES API.

Enabling the Exporting of Reputation Metrics for a Configuration Set

To start exporting reputation metrics for a configuration set, use the UpdateConfigurationSetReputationMetricsEnabled API operation. To access the Amazon SES API, we recommend using the AWS CLI or one of the AWS SDKs.

This procedure assumes that the AWS CLI is installed on your computer and properly configured. For more information about installing and configuring the AWS CLI, see the AWS Command Line Interface User Guide.
To enable the exporting of reputation metrics for a configuration set

- At the command line, type the following command: `aws ses update-configuration-set-reputation-metrics-enabled --configuration-set-name ConfigSet --enabled`

  Replace `ConfigSet` in the preceding command with the name of the configuration set for which you want to start exporting reputation metrics.

Disabling the Exporting of Reputation Metrics for a Configuration Set

You can also use the `UpdateConfigurationSetReputationMetricsEnabled` API operation to disable the exporting of reputation metrics for a configuration set.

To disable the exporting of reputation metrics for a configuration set

- At the command line, type the following command: `aws ses update-configuration-set-reputation-metrics-enabled --configuration-set-name ConfigSet --no-enabled`

  Replace `ConfigSet` in the preceding command with the name of the configuration set for which you want to disable the exporting of reputation metrics.
Monitoring Your Amazon SES Sending Activity

Amazon SES provides methods to monitor your sending activity. We recommend that you implement these methods so that you can keep track of important measures, such as your account's bounce, complaint and reject rates. Excessively high bounce and complaint rates may jeopardize your ability to send emails using Amazon SES.

You can also use these methods to measure the rates at which your customers engage with the emails you send. For example, these sending metrics can help you identify your overall open and clickthrough rates.

The metrics that you can measure using Amazon SES are referred to as *email sending events*. The email sending events that you can monitor are:

- **Sends** – Amazon SES accepted your email and attempted to deliver it to the recipient's email server.
- **Deliveries** – Amazon SES successfully delivered the email to the recipient's mail server.
- **Opens** – The recipient received the message and opened it in his or her email client.
- **Clicks** – The recipient clicked one or more links contained in the email.
- **Bounces** – The recipient's mail server permanently rejected the email. This event corresponds to hard bounces. Soft bounces are only included when Amazon SES fails to deliver the email after retrying for a period of time.
- **Complaints** – The recipient marked the email as spam.
- **Rejects** – Amazon SES accepted the email, determined that the email contained a virus, and rejected it.

You can monitor email sending events in three ways: using the console, using feedback notifications, and using event publishing. The monitoring method you choose depends on the type of event you want to monitor, the granularity and level of detail with which you want to monitor it, and where you want Amazon SES to publish the data. You might choose to use multiple monitoring methods. Characteristics of each method are listed in the following table.

<table>
<thead>
<tr>
<th>Monitoring Method</th>
<th>Events You Can Monitor</th>
<th>How to Access the Data</th>
<th>Level of Detail</th>
<th>Granularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon SES console</td>
<td>Deliveries and rejects</td>
<td>Sending Statistics page (p. 222) in Amazon SES console</td>
<td>Count only</td>
<td>Across entire AWS account</td>
</tr>
<tr>
<td>Amazon SES console</td>
<td>Bounce and complaint rates</td>
<td>Reputation Dashboard page (p. 329) in Amazon SES console</td>
<td>Calculated rates only</td>
<td>Across entire AWS account</td>
</tr>
<tr>
<td>Amazon SES API</td>
<td>Deliveries, bounces, complaints, and rejects</td>
<td>GetSendStatistics API operation</td>
<td>Count only</td>
<td>Across entire AWS account</td>
</tr>
</tbody>
</table>
### Monitoring Your Sending Statistics Using the Amazon SES Console

You can monitor the numbers of emails delivered from your account, as well as the number of messages that have been rejected, directly from the Amazon SES console. The delivery and rejection rates for your account are displayed on the Sending Statistics page.

You can find information about bounces and complaints on the Reputation Dashboard. For more information, see Monitoring Your Amazon SES Sender Reputation (p. 329).

---

#### Table: Monitoring Using the Console

<table>
<thead>
<tr>
<th>Monitoring Method</th>
<th>Events You Can Monitor</th>
<th>How to Access the Data</th>
<th>Level of Detail</th>
<th>Granularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon CloudWatch console</td>
<td>Sends, deliveries, opens, clicks, bounces, complaints, and rejects</td>
<td>CloudWatch console</td>
<td>Count only</td>
<td>Across entire AWS account</td>
</tr>
<tr>
<td>Feedback notifications</td>
<td>Deliveries, bounces, and complaints</td>
<td>Amazon SNS notification (deliveries, bounces, and complaints) or email (bounces and complaints only)</td>
<td>Details on each event</td>
<td>Across entire AWS account</td>
</tr>
<tr>
<td>Event publishing</td>
<td>Sends, deliveries, opens, clicks, bounces, complaints, and rejects</td>
<td>Amazon CloudWatch or Amazon Kinesis Firehose, or by Amazon SNS notification</td>
<td>Details on each event</td>
<td>Fine-grained (based on user-definable email characteristics)</td>
</tr>
</tbody>
</table>

**Note**

The metrics measured by email sending events may not align perfectly with your sending limits. This discrepancy can be caused by email bounces and rejections, or by using the Amazon SES inbox simulator. To find out how close you are to your sending limits, see Monitoring Your Sending Limits (p. 128).
Viewing Delivery and Rejection Metrics

To view delivery and rejection metrics

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation pane on the left side of the screen, choose Sending Statistics. Your usage statistics are shown under Your Amazon SES Metrics.
3. To view trend data for any metric, double-click the corresponding graph. When you double-click a graph, you can also change the analysis period.

Monitoring Your Usage Statistics Using the Amazon SES API

The Amazon SES API provides the GetSendStatistics operation, which returns information about your service usage. We recommend that you check your sending statistics regularly, so that you can make adjustments if needed.

When you call the GetSendStatistics operation, you receive a list of data points representing the last two weeks of your sending activity. Each data point in this list represents 15 minutes of activity and contains the following information for that period:

- The number of hard bounces
- The number of complaints
- The number of delivery attempts (corresponds to the number of emails you have sent)
- The number of rejected send attempts
- A timestamp for the analysis period

For a complete description of the GetSendStatistics operation, see the Amazon Simple Email Service API Reference.

In this section, you will find the following topics:

- the section called “Calling the GetSendStatistics API Operation Using the AWS CLI” (p. 223)
- the section called “Calling the GetSendStatistics Operation Programmatically” (p. 224)

Calling the GetSendStatistics API Operation Using the AWS CLI

The easiest way to call the GetSendStatistics API operation is to use the AWS Command Line Interface (AWS CLI).

To call the GetSendStatistics API operation using the AWS CLI

1. If you have not already done so, install the AWS CLI. For more information, see "Installing the AWS Command Line Interface” in the AWS Command Line Interface User Guide.
2. If you have not already done so, configure the AWS CLI to use your AWS credentials. For more information, see "Configuring the AWS CLI” in the AWS Command Line Interface User Guide.
3. At the command line, type `aws ses get-send-statistics`

If the AWS CLI is properly configured, you see a list of sending statistics in JSON format. Each JSON object includes aggregated sending statistics for a 15-minute period.

### Calling the GetSendStatistics Operation Programmatically

You can also call the `GetSendStatistics` operation using the AWS SDKs. This section includes code examples for the AWS SDKs for Go, PHP, Python, and Ruby. Choose one of the following links to view code examples for that language:

- Code example for the AWS SDK for Go (p. 224)
- Code example for the AWS SDK for PHP (p. 225)
- Code example for the AWS SDK for Python (Boto) (p. 225)
- Code example for the AWS SDK for Ruby (p. 226)

**Note**

These code examples assume that you have created an AWS shared credentials file that contains your AWS Access Key ID, your AWS Secret Access Key, and your preferred AWS Region. For more information, see Create a Shared Credentials File (p. 28).

### Calling GetSendStatistics Using the AWS SDK for Go

```go
package main

import (
    "fmt"
    //go get github.com/aws/aws-sdk-go/...
    "github.com/aws/aws-sdk-go/aws"
    "github.com/aws/aws-sdk-go/aws/session"
    "github.com/aws/aws-sdk-go/service/ses"
    "github.com/aws/aws-sdk-go/aws/awserr"
)

const (  
    // Replace us-west-2 with the AWS Region you're using for Amazon SES.  
    AwsRegion = "us-west-2"
)

func main() {
    // Create a new session and specify an AWS Region.  
    
    // Create an SES client in the session.  
    svc := ses.New(sess)
    input := &ses.GetSendStatisticsInput{}
    result, err := svc.GetSendStatistics(input)
    
    // Display error messages if they occur.  
    if err != nil {
        if aerr, ok := err.(awserr.Error); ok {
            if aerr, ok := err.(awserr.Error); ok {
                // Handle specific error types if needed.
            } else {
                // Handle general errors.
            }
        } else {
            // Handle unexpected error types.
        }
    }
    // Process the response.
}
```

224
Calling GetSendStatistics Using the AWS SDK for PHP

```php
<?php

// Replace path_to_sdk_inclusion with the path to the SDK as described in
define('REQUIRED_FILE', 'path_to_sdk_inclusion');

// Replace us-west-2 with the AWS Region you're using for Amazon SES.
define('REGION', 'us-west-2');

require REQUIRED_FILE;

use Aws\Ses\SesClient;

$client = SesClient::factory(array(
    'version' => 'latest',
    'region' => REGION
));

try {
    $result = $client->getSendStatistics([]);
    echo($result);
} catch (Exception $e) {
    echo($e->getMessage()."\n");
}
?>
```

Calling GetSendStatistics Using the AWS SDK for Python (Boto)

```python
import boto3 #pip install boto3
import json
from botocore.exceptions import ClientError

client = boto3.client('ses')

try:
    response = client.get_send_statistics()
except ClientError as e:
    print(e.response['Error']['Message'])
else:
    print(json.dumps(response, indent=4, sort_keys=True, default=str))
```
**Calling GetSendStatistics Using the AWS SDK for Ruby**

```ruby
require 'aws-sdk' # gem install aws-sdk
require 'json'

# Replace us-west-2 with the AWS Region you're using for Amazon SES.
awsregion = "us-west-2"

# Create a new SES resource and specify a region
ses = Aws::SES::Client.new(region: awsregion)

begin
  resp = ses.get_send_statistics({})
  puts JSON.pretty_generate(resp.to_h)
rescue Aws::SES::Errors::ServiceError => error
  puts error
end
```

---

**Monitoring Using Amazon SES Notifications**

Amazon SES can notify you of the status of your emails by email or through Amazon Simple Notification Service (Amazon SNS). Amazon SES supports the following three types of notifications:

- **Bounces** – The email is rejected by the recipient's ISP or rejected by Amazon SES because the email address is on the Amazon SES suppression list. For ISP bounces, Amazon SES reports only hard bounces and soft bounces that will no longer be retried by Amazon SES. In these cases, your recipient did not receive your email message, and Amazon SES will not try to resend it. Bounce notifications are available through email and Amazon SNS. You are notified of out-of-the-office (OOTO) messages through the same method as bounces, although they don’t count toward your bounce statistics. To see an example of an OOTO bounce notification, you can use the Amazon SES mailbox simulator. For more information, see Testing Amazon SES Email Sending (p. 162).

- **Complaints** – The email is accepted by the ISP and delivered to the recipient, but the recipient does not want the email and clicks a button such as "Mark as spam." If Amazon SES has a feedback loop set up with the ISP, Amazon SES will send you a complaint notification. Complaint notifications are available through email and Amazon SNS.

- **Deliveries** – Amazon SES successfully delivers the email to the recipient's mail server. This notification does not indicate that the actual recipient received the email because Amazon SES cannot control what happens to an email after the receiving mail server accepts it. Delivery notifications are available only through Amazon SNS.

You can set up notifications using the Amazon SES console or the Amazon SES API.

There are several important points to know about notifications:

- **Notification settings are configured on a per-verified-identity basis** – There is no global setting, and notification settings apply only to the verified identity in the AWS region in which you configured the settings.

- **Verified domain notification settings apply to all mail sent from email addresses in that domain except for email addresses that are also verified** – The configurations for email addresses are separate from the configuration for the domain, so changing the domain configuration will have no effect on the email address configuration. For example, if you verify only example.com and
configure its bounce notification settings, bounce notifications for email from \texttt{sender@example.com} will use those settings. However, if you verify both \texttt{example.com} and \texttt{sender@example.com}, \texttt{sender@example.com} will not use the bounce notification settings that are configured for \texttt{example.com}.

- **You must receive bounce and complaint notifications either by email or through Amazon SNS** – The default method is by email, through a feature called \textit{email feedback forwarding}. Delivery notifications are optional and available only through Amazon SNS.

- **If you choose to receive notifications for all three types of events, then you might receive multiple notifications for one email** – For example, the receiving mail server accepts the email (triggering a delivery notification), but the recipient marks the email as spam, triggering a complaint notification.

- **Before you start sending email, make sure that you set up a process to handle bounces and complaints** – Your process needs to monitor bounces and complaints and to remove those addresses from your mailing list. Excessive bounces and complaints put your Amazon SES account at risk of termination. You will need to analyze each bounce and complaint message that you receive to determine the cause. Bounces are usually caused by attempting to send to a nonexistent recipient; complaints arise when recipients indicate that they do not want to receive your message. In either case, we strongly recommend that you stop sending to these email addresses.

- **You can test notifications by using the Amazon SES mailbox simulator** – Emails that you send to the mailbox simulator do not affect your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 162).

The following sections describe the notification methods:

- To receive notifications by email (which applies to bounces and complaints only), see Amazon SES Notifications Through Email (p. 227).

- To receive notifications through Amazon SNS (which applies to all three notification types), see Amazon SES Notifications Through Amazon SNS (p. 229).

### Amazon SES Notifications Through Email

Amazon SES can notify you of your bounces and complaints through email using a process called \textit{email feedback forwarding} or through Amazon Simple Notification Service (Amazon SNS). This topic is about receiving notifications by email, which is the default setting. For information about setting up notifications through Amazon SNS, see Amazon SES Notifications Through Amazon SNS (p. 229). Unlike bounce and complaint notifications, delivery notifications are available only through Amazon SNS.

**Important**

For several important points about notifications, see Monitoring Using Amazon SES Notifications (p. 226).

The following sections describe how to receive bounce and complaint notifications through email:

- To enable bounce and complaint notifications by email, see Enabling Email Feedback Forwarding (p. 227).

- To disable bounce and complaint notifications by email, see Disabling Email Feedback Forwarding (p. 228).

- To learn the email address to which bounce and complaint notifications are sent, see Email Feedback Forwarding Destination (p. 228).

### Enabling Email Feedback Forwarding

Email feedback forwarding is enabled by default. If you previously disabled it, you can enable it by following the procedures in this section.
To enable bounce and complaint forwarding through email using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation pane, under Identity Management, choose Email Addresses if you want to configure bounce and complaint notifications for an email address, or choose Domains if you want to configure bounce and complaint notifications for a domain.
3. In the list of verified senders, choose the email address or domain for which you want to configure bounce and complaint notifications.
4. In the details pane, expand the Notifications section.
5. Choose Edit Configuration.
6. Under Email Feedback Forwarding, choose Enabled.
   Note
   Changes made to your settings on this page might take a few minutes to take effect.

You can also enable bounce and complaint notifications through email by using the SetIdentityFeedbackForwardingEnabled API operation.

Disabling Email Feedback Forwarding

You must receive bounce and complaint notifications through either Amazon SNS or email feedback forwarding, so you can disable email feedback forwarding only if you select an Amazon SNS topic for both bounce and complaint notifications. If you selected an Amazon SNS topic for bounces and complaints, you can disable email feedback forwarding by using the following procedure.

To disable bounce and complaint forwarding through email using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation pane, under Identity Management, choose Email Addresses if you want to configure bounce and complaint notifications for an email address, or choose Domains if you want to configure bounce and complaint notifications for a domain.
3. In the list of verified senders, choose the email address or domain for which you want to configure bounce and complaint notifications.
4. In the details pane, expand the Notifications section.
5. Choose Edit Configuration.
6. In the Edit Notification Configuration dialog box, ensure that you have selected an Amazon SNS topic for both bounces and complaints. Otherwise, you will not be able to disable email feedback forwarding in the next step.
7. Under Email Feedback Forwarding, choose Disabled.
8. Choose Save Config to save your notification configuration.
   Note
   Changes made to your settings on this page might take a few minutes to take effect.

You can also disable bounce and complaint notifications through email by using the SetIdentityFeedbackForwardingEnabled API operation.

Email Feedback Forwarding Destination

When you receive notifications by email, Amazon SES rewrites the From header and sends the notification to you. The address to which Amazon SES forwards the notification depends on how you sent the original message.
If you used the SMTP interface to send the message, then notifications go to the address specified in the MAIL FROM command, which overrides any Return-Path headers specified in the SMTP DATA.

If you used the SendEmail API operation to send the message, then the notifications are delivered as follows:

- If you specified the optional ReturnPath parameter of SendEmail, then notifications go to that address.
- Otherwise, notifications go to the address specified in the required Source parameter of SendEmail, which populates the From header of the message.

If you used the SendRawEmail API operation to send the message, then the notifications are delivered as follows:

- If you specified the optional Source parameter of SendRawEmail, then notifications go to that address, overriding any Return-Path headers specified in the raw message.
- Otherwise, if the Return-Path header was specified in the raw message, then notifications go to that address.
- Otherwise, notifications go to the address in the From header of the raw message.

**Important**
Regardless of whether you use the SMTP interface, SendEmail API, or SendRawEmail API, Amazon SES overwrites any Return-Path headers that you provide.

### Amazon SES Notifications Through Amazon SNS

You can use Amazon Simple Notification Service (Amazon SNS) to receive notifications about bounces, complaints, and deliveries for emails that you send through Amazon SES. Amazon SNS notifications are in JavaScript Object Notation (JSON) format, which enables you to process them programmatically.

If you configure Amazon SNS notifications for both bounces and complaints, you can disable receiving bounce and complaint notifications by email entirely. For information about receiving bounce and complaint notifications by email, see Amazon SES Notifications Through Email (p. 227). Delivery notifications are available only through Amazon SNS.

**Important**
For several important points about notifications, see Monitoring Using Amazon SES Notifications (p. 226).

For information about Amazon SES bounce, complaint, and delivery notifications through Amazon SNS, see the following sections:

- To set up notifications using the Amazon SES console or the Amazon SES API, see Configuring Amazon SNS Notifications for Amazon SES (p. 229).
- For a description of the contents of a notification, see Amazon SNS Notification Contents for Amazon SES (p. 231).
- For examples of bounce, complaint, and delivery notifications, see Amazon SNS Notification Examples for Amazon SES (p. 240).

### Configuring Amazon SNS Notifications for Amazon SES

Amazon SES can notify you of your bounces, complaints, and deliveries through Amazon Simple Notification Service (Amazon SNS).

You can configure notifications in the Amazon SES console, or by using the Amazon SES API.
Prerequisites

Complete the following steps before you set up Amazon SNS notifications in Amazon SES:

1. Create a topic in Amazon SNS. For more information, see Create a Topic in the Amazon Simple Notification Service Developer Guide.
2. Subscribe at least one endpoint to the topic. For example, if you want to receive notifications by text message, subscribe an SMS endpoint (that is, a mobile phone number) to the topic. To receive notifications by email, subscribe an email endpoint (an email address) to the topic.

   For more information, see Subscribe to a Topic in the Amazon Simple Notification Service Developer Guide.

Configuring Notifications Using the Amazon SES Console

To configure notifications using the Amazon SES console

1. Open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation pane, under Identity Management, choose Domains or Email Addresses.
3. In the list of verified senders, choose the email address or domain you want to configure notifications for.
   
   Important
   
   Verified domain notification settings apply to all mail sent from email addresses in that domain except for email addresses that are also verified.

5. Under SNS Topic Configuration, make the following changes to the Amazon SNS topic configuration:

   a. Choose the Amazon SNS topics you want to use to receive notifications. You can publish multiple event type notifications to the same Amazon SNS topic or to different Amazon SNS topics.
      
      Important
      
      The Amazon SNS topics you use for bounce, complaint, and delivery notifications must be in the same AWS Region where you use Amazon SES. Additionally, the topics you select must be subscribed to by one or more endpoints. For example, if you want to have notifications sent to an email address, you must subscribe an email endpoint to the topic. For more information, see Subscribe to a Topic in the Amazon Simple Notification Service Developer Guide.

      If you want to use an Amazon SNS topic that you do not own, you must configure your AWS Identity and Access Management (IAM) policy to allow publishing from the Amazon Resource Name (ARN) of the Amazon SNS topic.

   b. If you want the Amazon SNS notifications to contain the original headers of the emails you pass to Amazon SES, choose Include original headers. This option is only available if you have assigned an Amazon SNS topic to the associated notification type. For information about the contents of the original email headers, see the mail object in Amazon SNS Notification Contents (p. 231).
6. If you choose Amazon SNS topics for both bounces and complaints, you can disable email notifications entirely. To disable email notifications for bounces and complaints, under Email Feedback Forwarding, choose Disable. Delivery notifications are available only through Amazon SNS.

7. Choose Save Config. The changes you made to your notification settings might take a few minutes to take effect.

After you configure your settings, you will start receiving bounce, complaint, and/or delivery notifications to your Amazon SNS topic(s). These notifications will be in JavaScript Object Notation (JSON) format and will follow the structure described in Amazon SNS Notification Contents (p. 231).

You will be charged standard Amazon SNS rates for bounce, complaint, and delivery notifications. For more information, see the Amazon SNS pricing page.

Note
If an attempt to publish to your Amazon SNS topic fails because the topic has been deleted or your AWS account no longer has permissions to publish to it, the Amazon SES configuration for that topic for the sending identity will be deleted, bounce and complaint notifications through email will be re-enabled for that identity, and you will be notified of the change through email.

If you have multiple identities configured to use that topic, each identity will have its topic configuration changed when each identity experiences a failure to publish to that topic.

Configuring Notifications Using the Amazon SES API

You can also configure bounce, complaint, and delivery notifications by using the Amazon SES API. Use the following operations to configure notifications:

- SetIdentityNotificationTopic
- SetIdentityFeedbackForwardingEnabled
- GetIdentityNotificationAttributes
- SetIdentityHeadersInNotificationsEnabled

You can use these API actions to write a customized front-end application for notifications. For a complete description of the API actions related to notifications, see the Amazon Simple Email Service API Reference.

Amazon SNS Notification Contents for Amazon SES

Bounce, complaint, and delivery notifications are published to Amazon Simple Notification Service (Amazon SNS) topics in JavaScript Object Notation (JSON) format. The top-level JSON object contains a notificationType string, a mail object, and either a bounce object, a complaint object, or a delivery object.

See the following sections for descriptions of the different types of objects:

- Top-level JSON object (p. 232)
- mail object (p. 232)
- bounce object (p. 235)
- complaint object (p. 238)
- delivery object (p. 239)

The following are some important notes about the contents of Amazon SNS notifications for Amazon SES:
• For a given notification type, you might receive one Amazon SNS notification for multiple recipients, or you might receive a single Amazon SNS notification per recipient. Your code should be able to parse the Amazon SNS notification and handle both cases; Amazon SES does not make ordering or batching guarantees for notifications sent through Amazon SNS. However, different Amazon SNS notification types (for example, bounces and complaints) will never be combined into a single notification.

• You might receive multiple types of Amazon SNS notifications for one recipient. For example, the receiving mail server might accept the email (triggering a delivery notification), but after processing the email, the receiving mail server might determine that the email actually results in a bounce (triggering a bounce notification). However, these will always be separate notifications because they are different notification types.

• Amazon SES reserves the right to add additional fields to the notifications. As such, applications that parse these notifications must be flexible enough to handle unknown fields.

• Amazon SES overwrites the headers of the message when it sends the email. You can retrieve the headers of the original message from the headers and commonHeaders fields of the mail object.

Top-Level JSON Object

The top-level JSON object in an Amazon SES notification contains the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>notificationType</td>
<td>A string that holds the type of notification represented by the JSON object. Possible values are Bounce, Complaint, or Delivery.</td>
</tr>
<tr>
<td>mail</td>
<td>A JSON object that contains information about the original mail to which the notification pertains. For more information, see Mail Object (p. 232).</td>
</tr>
<tr>
<td>bounce</td>
<td>This field is present only if the notificationType is Bounce and contains a JSON object that holds information about the bounce. For more information, see Bounce Object (p. 235).</td>
</tr>
<tr>
<td>complaint</td>
<td>This field is present only if the notificationType is Complaint and contains a JSON object that holds information about the complaint. For more information, see Complaint Object (p. 238).</td>
</tr>
<tr>
<td>delivery</td>
<td>This field is present only if the notificationType is Delivery and contains a JSON object that holds information about the delivery. For more information, see Delivery Object (p. 239).</td>
</tr>
</tbody>
</table>

Mail Object

Each bounce, complaint, or delivery notification contains information about the original email in the mail object. The JSON object that contains information about a mail object has the following fields.
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp</td>
<td>The time at which the original message was sent (in ISO8601 format).</td>
</tr>
<tr>
<td>messageId</td>
<td>A unique ID that Amazon SES assigned to the message. Amazon SES returned this value to you when you sent the message.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>This message ID was assigned by Amazon SES. You can find the message ID of the original email in the headers and commonHeaders fields of the mail object.</td>
</tr>
<tr>
<td>source</td>
<td>The email address from which the original message was sent (the envelope MAIL FROM address).</td>
</tr>
<tr>
<td>sourceArn</td>
<td>The Amazon Resource Name (ARN) of the identity that was used to send the email. In the case of sending authorization, the sourceArn is the ARN of the identity that the identity owner authorized the delegate sender to use to send the email. For more information about sending authorization, see Using Sending Authorization (p. 132).</td>
</tr>
<tr>
<td>sourceIp</td>
<td>The originating public IP address of the client that performed the email sending request to Amazon SES.</td>
</tr>
<tr>
<td>sendingAccountId</td>
<td>The AWS account ID of the account that was used to send the email. In the case of sending authorization, the sendingAccountId is the delegate sender's account ID.</td>
</tr>
<tr>
<td>destination</td>
<td>A list of email addresses that were recipients of the original mail.</td>
</tr>
<tr>
<td>headersTruncated</td>
<td>(Only present if the notification settings (p. 229) include the original email headers.) A string that specifies whether the headers are truncated in the notification, which occurs if the headers are larger than 10 KB. Possible values are true and false.</td>
</tr>
<tr>
<td>headers</td>
<td>(Only present if the notification settings include the original email headers.) A list of the email's original headers. Each header in the list has a name field and a value field.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>Any message ID within the headers field is from the original message that you passed to Amazon SES. The message ID that Amazon SES subsequently assigned to the message is in the messageId field of the mail object.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>commonHeaders</td>
<td>(Only present if the notification settings include the original email headers.) A list of the email's original, commonly used headers. Each header in the list has a name field and a value field.</td>
</tr>
</tbody>
</table>

**Note**

Any message ID within the commonHeaders field is from the original message that you passed to Amazon SES. The message ID that Amazon SES subsequently assigned to the message is in the messageId field of the mail object.

The following is an example of a mail object that includes the original email headers. When this notification type is not configured to include the original email headers, the mail object does not include the headersTruncated, headers, and commonHeaders fields.

```json
{
  "timestamp": "2016-01-27T14:05:45 +0000",
  "messageId": "0000001378603177f-97a543397-8e8d-42ae-af10-0186f34d6ee-000000",
  "source": "john@example.com",
  "sourceIp": "127.0.0.1",
  "sendingAccount": "123456789012",
  "destination": ["jane@example.com"],
  "headersTruncated": false,
  "headers": [
    {
      "name": "From",
      "value": "John Doe <john@example.com>"
    },
    {
      "name": "To",
      "value": "Jane Doe <jane@example.com>"
    },
    {
      "name": "Message-ID",
      "value": "custom-message-ID"
    },
    {
      "name": "Subject",
      "value": "Hello"
    },
    {
      "name": "Content-Type",
      "value": "text/plain; charset=UTF-8"
    },
    {
      "name": "Content-Transfer-Encoding",
      "value": "base64"
    },
    {
      "name": "Date",
      "value": "Wed, 27 Jan 2016 14:05:45 +0000"
    }
  ]
}
```
Bounce Object

The JSON object that contains information about bounces will always have the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bounceType</td>
<td>The type of bounce, as determined by Amazon SES. For more information, see Bounce Types (p. 237).</td>
</tr>
<tr>
<td>bounceSubType</td>
<td>The subtype of the bounce, as determined by Amazon SES. For more information, see Bounce Types (p. 237).</td>
</tr>
<tr>
<td>bouncedRecipients</td>
<td>A list that contains information about the recipients of the original mail that bounced. For more information, see Bounced Recipients (p. 236).</td>
</tr>
<tr>
<td>timestamp</td>
<td>The date and time at which the bounce was sent (in ISO8601 format). Note that this is the time at which the notification was sent by the ISP, and not the time at which it was received by Amazon SES.</td>
</tr>
<tr>
<td>feedbackId</td>
<td>A unique ID for the bounce.</td>
</tr>
</tbody>
</table>

If Amazon SES was able to contact the remote Message Transfer Authority (MTA), the following field will also be present.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>remoteMtaIp</td>
<td>The IP address of the MTA to which Amazon SES attempted to deliver the email.</td>
</tr>
</tbody>
</table>

If a delivery status notification (DSN) was attached to the bounce, the following field may also be present.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reportingMTA</td>
<td>The value of the Reporting-MTA field from the DSN. This is the value of the MTA that attempted to perform the delivery, relay, or gateway operation described in the DSN.</td>
</tr>
</tbody>
</table>
The following is an example of a bounce object.

```
{
  "bounceType": "Permanent",
  "bounceSubType": "General",
  "bouncedRecipients": [
    {
      "status": "5.0.0",
      "action": "failed",
      "diagnosticCode": "smtp; 550 user unknown",
      "emailAddress": "recipient1@example.com"
    },
    {
      "status": "4.0.0",
      "action": "delayed",
      "emailAddress": "recipient2@example.com"
    }
  ],
  "reportingMTA": "example.com",
  "timestamp": "2012-05-25T14:59:38.605Z",
  "feedbackId": "000001378603176d-5a4b5ad9-6f30-4198-a8c3-b1eb0c270a1d-000000",
  "remoteMtaIp": "127.0.2.0"
}
```

**Bounced Recipients**

A bounce notification may pertain to a single recipient or to multiple recipients. The `bouncedRecipients` field holds a list of objects—one per recipient to whom the bounce notification pertains—and will always contain the following field.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>emailAddress</td>
<td>The email address of the recipient. If a DSN is available, this is the value of the Final-Recipient field from the DSN.</td>
</tr>
</tbody>
</table>

Optionally, if a DSN is attached to the bounce, the following fields may also be present.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>The value of the Action field from the DSN. This indicates the action performed by the Reporting-MTA as a result of its attempt to deliver the message to this recipient.</td>
</tr>
<tr>
<td>status</td>
<td>The value of the Status field from the DSN. This is the per-recipient transport-independent status code that indicates the delivery status of the message.</td>
</tr>
<tr>
<td>diagnosticCode</td>
<td>The status code issued by the reporting MTA. This is the value of the Diagnostic-Code field from the DSN. This field may be absent in the DSN (and therefore also absent in the JSON).</td>
</tr>
</tbody>
</table>

The following is an example of an object that might be in the `bouncedRecipients` list.
Bounce Types

The following bounce types are available. We recommend that you remove the email addresses that have returned bounces marked Permanent from your mailing list, as we do not believe that you will be able to successfully send to them in the future. Transient bounces are sent to you when all retry attempts have been exhausted and will no longer be retried. You may be able to successfully resend to an address that initially resulted in a Transient bounce.

Note

Amazon SES only reports hard bounces and soft bounces that will no longer be retried by Amazon SES. In other words, your recipient did not receive your email message, and Amazon SES will not try to resend it.

<table>
<thead>
<tr>
<th>bounceType</th>
<th>bounceSubType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undetermined</td>
<td>Undetermined</td>
<td>Amazon SES was unable to determine a specific bounce reason.</td>
</tr>
<tr>
<td>Permanent</td>
<td>General</td>
<td>Amazon SES received a general hard bounce and recommends that you remove the recipient's email address from your mailing list.</td>
</tr>
<tr>
<td>Permanent</td>
<td>NoEmail</td>
<td>Amazon SES received a permanent hard bounce because the target email address does not exist. It is recommended that you remove that recipient from your mailing list.</td>
</tr>
<tr>
<td>Permanent</td>
<td>Suppressed</td>
<td>Amazon SES has suppressed sending to this address because it has a recent history of bouncing as an invalid address. For information about how to remove an address from the suppression list, see Removing an Email Address from the Amazon SES Suppression List (p. 428).</td>
</tr>
<tr>
<td>Transient</td>
<td>General</td>
<td>Amazon SES received a general bounce. You may be able to successfully retry sending to that recipient in the future.</td>
</tr>
<tr>
<td>Transient</td>
<td>MailboxFull</td>
<td>Amazon SES received a mailbox full bounce. You may be able to successfully retry sending to that recipient in the future.</td>
</tr>
<tr>
<td>Transient</td>
<td>MessageTooLarge</td>
<td>Amazon SES received a message too large bounce. You may be able to successfully retry sending to that recipient if you reduce the message size.</td>
</tr>
<tr>
<td>Transient</td>
<td>ContentRejected</td>
<td>Amazon SES received a content rejected bounce. You may be able to successfully retry sending to that recipient if you change the message content.</td>
</tr>
<tr>
<td>Transient</td>
<td>AttachmentRejected</td>
<td>Amazon SES received an attachment rejected bounce. You may be able to successfully retry</td>
</tr>
<tr>
<td>bounceType</td>
<td>bounceSubType</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sending to that recipient if you remove or change the attachment.</td>
</tr>
</tbody>
</table>

**Complaint Object**

The JSON object that contains information about complaints has the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>complainedRecipients</td>
<td>A list that contains information about recipients that may have been responsible for the complaint. For more information, see Complained Recipients (p. 239).</td>
</tr>
<tr>
<td>timestamp</td>
<td>The date and time at which the complaint was sent (in ISO8601 format). Note that this is the time at which the notification was sent by the ISP, and not the time at which it was received by Amazon SES.</td>
</tr>
<tr>
<td>feedbackId</td>
<td>A unique ID for the complaint.</td>
</tr>
</tbody>
</table>

Further, if a feedback report is attached to the complaint, the following fields may be present.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userAgent</td>
<td>The value of the User-Agent field from the feedback report. This indicates the name and version of the system that generated the report.</td>
</tr>
<tr>
<td>complaintFeedbackType</td>
<td>The value of the Feedback-Type field from the feedback report received from the ISP. This contains the type of feedback.</td>
</tr>
<tr>
<td>arrivalDate</td>
<td>The value of the Arrival-Date or Received-Date field from the feedback report (in ISO8601 format). This field may be absent in the report (and therefore also absent in the JSON).</td>
</tr>
</tbody>
</table>

The following is an example of a complaint object.

```json
{
    "userAgent":"AnyCompany Feedback Loop (V0.01),
    "complainedRecipients":[
        {
            "emailAddress":"recipient1@example.com"
        }
    ],
    "complaintFeedbackType":"abuse",
    "arrivalDate":"2009-12-03T04:24:21.000-05:00",
    "feedbackId":"000001378603177f-18c07c78-fa81-4a58-9dd1-fedc3cb8f49a-000000"
}
```
Complained Recipients

The `complainedRecipients` field contains a list of recipients that may have submitted the complaint.

**Important**
Since most ISPs redact the email address of the recipient who submitted the complaint from their complaint notification, this list contains information about recipients who might have sent the complaint, based on the recipients of the original message and the ISP from which we received the complaint. Amazon SES performs a lookup against the original message to determine this recipient list.

JSON objects in this list contain the following field:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>emailAddress</code></td>
<td>The email address of the recipient.</td>
</tr>
</tbody>
</table>

The following is an example of a Complained Recipient object.

```json
{ "emailAddress": "recipient1@example.com" }
```

**Note**
Because of this behavior, you can be more certain that you know which email address complained about your message if you limit your sending to one message per recipient (rather than sending one message with 30 different email addresses in the bcc line).

Complaint Types

You may see the following complaint types in the `complaintFeedbackType` field as assigned by the reporting ISP, according to the Internet Assigned Numbers Authority website:

- `abuse`—Indicates unsolicited email or some other kind of email abuse.
- `auth-failure`—Email authentication failure report.
- `fraud`—Indicates some kind of fraud or phishing activity.
- `not-spam`—Indicates that the entity providing the report does not consider the message to be spam. This may be used to correct a message that was incorrectly tagged or categorized as spam.
- `other`—Indicates any other feedback that does not fit into other registered types.
- `virus`—Reports that a virus is found in the originating message.

Delivery Object

The JSON object that contains information about deliveries will always have the following fields:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>timestamp</code></td>
<td>The time Amazon SES delivered the email to the recipient's mail server (in ISO8601 format).</td>
</tr>
<tr>
<td><code>processingTimeMillis</code></td>
<td>The time in milliseconds between when Amazon SES accepted the request from the sender to passing the message to the recipient's mail server.</td>
</tr>
</tbody>
</table>
**Field Name** | **Description**
--- | ---
recipients | A list of the intended recipients of the email to which the delivery notification applies.
smtpResponse | The SMTP response message of the remote ISP that accepted the email from Amazon SES. This message will vary by email, by receiving mail server, and by receiving ISP.
reportingMTA | The host name of the Amazon SES mail server that sent the mail.
remoteMtaIp | The IP address of the MTA to which Amazon SES delivered the email.

The following is an example of a delivery object.

```
{
  "processingTimeMillis":546,
  "recipients":["success@simulator.amazonses.com"],
  "smtpResponse":"250 ok: Message 64111812 accepted",
  "reportingMTA":"a8-70.smtp-out.amazonses.com",
  "remoteMtaIp":"127.0.2.0"
}
```

### Amazon SNS Notification Examples for Amazon SES

The following sections provide examples of the three types of notifications:

- For bounce notification examples, see Amazon SNS Bounce Notification Examples (p. 240).
- For complaint notification examples, see Amazon SNS Complaint Notification Examples (p. 243).
- For delivery notification examples, see Amazon SNS Delivery Notification Example (p. 245).

### Amazon SNS Bounce Notification Examples

This section contains examples of bounce notifications with and without a Delivery Status Notification (DSN) provided by the email receiver that sent the feedback.

#### Bounce Notification With a DSN

The following is an example of a bounce notification that contains a DSN and the original email headers. When bounce notifications are not configured to include the original email headers, the mail object within the notifications does not include the headersTruncated, headers, and commonHeaders fields.

```
{
  "notificationType":"Bounce",
  "bounce":{
    "bounceType":"Permanent",
    "reportingMTA":"dns; email.example.com",
    "smtpResponse":"250 ok: Message 64111812 accepted",
    "bouncedRecipients": [
      {
        "emailAddress":"jane@example.com",
        "status":"5.1.1",
        "action":"failed",
        "diagnosticCode":"smtp; 550 5.1.1 <jane@example.com>... User"
      }
    ]
  }
}
```
{ "bounceSubType": "General", "timestamp": "2016-01-27T14:59:38.237Z", "feedbackId": "00000138111222aa-33322211-cccc-cccc-cccc-dddwww068a-000000", "remoteMtaIp": "127.0.2.0" },
"mail": {
  "timestamp": "2016-01-27T14:59:38.237Z",
  "source": "john@example.com",
  "sourceIp": "127.0.3.0",
  "sendingAccountId": "123456789012",
  "messageId": "00000138111222aa-33322211-cccc-cccc-cccc-dddwww0680-000000",
  "destination": [
    "jane@example.com",
    "mary@example.com",
    "richard@example.com"
  ],
  "headersTruncated": false,
  "headers": [
    { "name": "From", "value": ""John Doe" <john@example.com>
    },
    { "name": "To", "value": ""Jane Doe" <jane@example.com>, "Mary Doe" <mary@example.com>, "Richard Doe" <richard@example.com>
    },
    { "name": "Message-ID", "value": "custom-message-ID"
    },
    { "name": "Subject", "value": "Hello"
    },
    { "name": "Content-Type", "value": "text/plain; charset=\"UTF-8\"
    },
    { "name": "Content-Transfer-Encoding", "value": "base64"
    },
    { "name": "Date", "value": "Wed, 27 Jan 2016 14:05:45 +0000"
    }
  ],
  "commonHeaders": {
    "from": [
      "John Doe <john@example.com>"
    ],
    "date": "Wed, 27 Jan 2016 14:05:45 +0000",
    "to": [
      "Jane Doe <jane@example.com>, Mary Doe <mary@example.com>, Richard Doe <richard@example.com>
    ],
    "messageId": "custom-message-ID",
    "subject": "Hello"
  }
}

241
Bounce Notification Without a DSN

The following is an example of a bounce notification that includes the original email headers but does not include a DSN. When bounce notifications are not configured to include the original email headers, the mail object within the notifications does not include the headersTruncated, headers, and commonHeaders fields.

```json
{
    "notificationType": "Bounce",
    "bounce": {
        "bounceType": "Permanent",
        "bounceSubType": "General",
        "bouncedRecipients": [
            {
                "emailAddress": "jane@example.com"
            },
            {
                "emailAddress": "richard@example.com"
            }
        ],
        "timestamp": "2016-01-27T14:59:38.237Z",
        "feedbackId": "00000137860315fd-869464a4-8680-4114-98d3-716fe35851f9-000000",
        "remoteMtaIp": "127.0.2.0"
    },
    "mail": {
        "timestamp": "2016-01-27T14:59:38.237Z",
        "messageId": "00000137860315fd-34208509-5b74-41f3-95c5-22c1edc3c924-000000",
        "sourceIp": "127.0.3.0",
        "sendingAccountId": "123456789012",
        "destination": [
            "jane@example.com",
            "mary@example.com",
            "richard@example.com"
        ],
        "headersTruncated": false,
        "headers": [
            {
                "name": "From",
                "value": ""John Doe" <john@example.com>"
            },
            {
                "name": "To",
                "value": ""Jane Doe" <jane@example.com>, "Mary Doe" <mary@example.com>, "Richard Doe" <richard@example.com>"
            },
            {
                "name": "Message-ID",
                "value": "custom-message-ID"
            },
            {
                "name": "Subject",
                "value": "Hello"
            },
            {
                "name": "Content-Type",
                "value": "text/plain; charset="UTF-8"
            },
            {
                "name": "Content-Transfer-Encoding",
                "value": "base64"
            },
            {
                "name": "Date",
                "value": "2016-01-27T14:59:38.237Z"
            }
        ]
    }
}
```
Amazon SNS Complaint Notification Examples

This section contains examples of complaint notifications with and without a feedback report provided by the email receiver that sent the feedback.

Complaint Notification With a Feedback Report

The following is an example of a complaint notification that contains a feedback report and the original email headers. When complaint notifications are not configured to include the original email headers, the mail object within the notifications does not include the headersTruncated, headers, and commonHeaders fields.

```json
{
    "notificationType":"Complaint",
    "complaint":{
        "userAgent":"AnyCompany Feedback Loop (V0.01)",
        "complainedRecipients":{
            "emailAddress":"richard@example.com"
        },
        "complaintFeedbackType":"abuse",
        "arrivalDate":"2016-01-27T14:59:38.237Z",
        "timestamp":"2016-01-27T14:59:38.237Z",
        "feedbackId":"000001378603177f-18c07c78-fa81-4a58-9dd1-fedc3cb8f49a-000000"
    },
    "mail":{
        "timestamp":"2016-01-27T14:59:38.237Z",
        "messageId":"000001378603177f-7a5433e7-8ed8-42ae-af10-f0181f34d6ee-000000",
        "source":"john@example.com",
        "sourceIp": "127.0.0.1",
        "sendingAccountId":"123456789012",
        "destination":{
            "jane@example.com",
            "mary@example.com",
            "richard@example.com"
        },
        "headersTruncated":false,
        "headers":{
            "name":"From",
            "value":""}\n        },
        "name":"To",
        "value":""\n    }
}
```
Complaint Notification Without a Feedback Report

The following is an example of a complaint notification that includes the original email headers but does not include a feedback report. When complaint notifications are not configured to include the original email headers, the mail object within the notifications does not include the headersTruncated, headers, and commonHeaders fields.

```json
{
  "notificationType": "Complaint",
  "complaint": {
    "complainedRecipients": [
      {
        "emailAddress": "richard@example.com"
      }
    ],
    "timestamp": "2016-01-27T14:59:38.237Z",
    "feedbackId": "0000013786031775-fea503bc-7497-49e1-881b-a0379bb037d3-000000"
  },
  "mail": {
    "timestamp": "2016-01-27T14:59:38.237Z",
    "messageId": "0000013786031775-163e3910-53eb-4c8e-a04a-f29debf88a84-000000",
    "source": "john@example.com",
    "sourceIp": "127.0.0.1"
  }
}
```
Amazon SNS Delivery Notification Example

The following is an example of a delivery notification that includes the original email headers. When delivery notifications are not configured to include the original email headers, the `mail` object within the notifications does not include the `headersTruncated`, `headers`, and `commonHeaders` fields.

```json
{
  "notificationType": "Delivery",
  "mail": {
    "timestamp": "2016-01-27T14:59:38.237Z",
    "messageId": "0000014644fe5ef6-9a483358-9170-4cb4-a269-f5dcdf415321-000000",
    "source": "john@example.com",
    "headers": [
      {
        "name": "From",
        "value": "John Doe <john@example.com>",
      },
      {
        "name": "To",
        "value": "Jane Doe <jane@example.com>, Mary Doe <mary@example.com>, Richard Doe <richard@example.com>",
      },
      {
        "name": "Message-ID",
        "value": "custom-message-ID"
      },
      {
        "name": "Subject",
        "value": "Hello"
      },
      {
        "name": "Content-Type",
        "value": "text/plain; charset=\"UTF-8\"
      },
      {
        "name": "Content-Transfer-Encoding",
        "value": "base64"
      },
      {
        "name": "Date",
        "value": "Wed, 27 Jan 2016 14:05:45 +0000"
      }
    ],
    "commonHeaders": {
      "from": [
        "John Doe <john@example.com>"
      ],
      "date": "Wed, 27 Jan 2016 14:05:45 +0000",
      "to": [
        "Jane Doe <jane@example.com>, Mary Doe <mary@example.com>, Richard Doe <richard@example.com>"
      ],
      "messageId": "custom-message-ID",
      "subject": "Hello"
    }
  }
}
```
"sourceIp": "127.0.3.0",
"sendingAccountId": "123456789012",
"destination": [
  "jane@example.com"
],
"headersTruncated": false,
"headers": [
  {
    "name": "From",
    "value": ""John Doe" <john@example.com>"
  },
  {
    "name": "To",
    "value": ""Jane Doe" <jane@example.com>"
  },
  {
    "name": "Message-ID",
    "value": "custom-message-ID"
  },
  {
    "name": "Subject",
    "value": "Hello"
  },
  {
    "name": "Content-Type",
    "value": "text/plain; charset="UTF-8""
  },
  {
    "name": "Content-Transfer-Encoding",
    "value": "base64"
  },
  {
    "name": "Date",
    "value": "Wed, 27 Jan 2016 14:58:45 +0000"
  }
],
"commonHeaders": {
  "from": [
    "John Doe <john@example.com>
  ],
  "date": "Wed, 27 Jan 2016 14:58:45 +0000",
  "to": [
    "Jane Doe <jane@example.com>
  ],
  "messageId": "custom-message-ID",
  "subject": "Hello"
},
"delivery": {
  "timestamp": "2016-01-27T14:59:38.237Z",
  "recipients": ["jane@example.com"],
  "processingTimeMillis": 546,
  "reportingMTA": "a8-70.smtp-out.amazonaws.com",
  "smtpResponse": "250 ok: Message 64111812 accepted",
  "remoteMtaIp": "127.0.2.0"
}
Monitoring Using Amazon SES Event Publishing

To enable you to track your email sending at a granular level, you can set up Amazon SES to publish email sending events to Amazon CloudWatch or Amazon Kinesis Firehose based on fine-grained email characteristics that you define. For example, you can categorize your emails by purpose (transactional versus marketing), product details, the recipient's "From" domain, and so on.

You can track several types of email sending events, including sends, deliveries, opens, clicks, bounces, complaints, and rejections. This information can be useful for operational and analytical purposes. For example, you can publish your email sending data to CloudWatch and create dashboards that track the performance of your email campaigns.

How Event Publishing Works

To use event publishing, you first set up one or more configuration sets. A configuration set specifies where to publish your events and which events to publish. Then, each time you send an email, you provide the name of the configuration set and one or more message tags, in the form of name/value pairs, to categorize the email. For example, if you advertise books, you could name a message tag genre, and assign a value of sci-fi or western, when you send an email for the associated campaign. Depending on which email sending interface you use, you either provide the message tag as a parameter to the API call or as an Amazon SES-specific email header. For more information about configuration sets, see Using Amazon SES Configuration Sets (p. 215).

In addition to the message tags that you specify, Amazon SES also adds auto-tags to the messages you send. You do not need to perform any additional steps to use auto-tags.

The following table lists the auto-tags that are automatically applied to messages you send using Amazon SES.

Amazon SES Auto-Tags

<table>
<thead>
<tr>
<th>Auto-tag name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ses:configuration-set</td>
<td>The name of the Configuration Set associated with the email.</td>
</tr>
<tr>
<td>ses:caller-identity</td>
<td>The IAM identity of the Amazon SES user who sent the email.</td>
</tr>
<tr>
<td>ses:from-domain</td>
<td>The domain of the &quot;From&quot; address.</td>
</tr>
<tr>
<td>ses:source-ip</td>
<td>The IP address that the caller used to send the email.</td>
</tr>
<tr>
<td>ses:outgoing-ip</td>
<td>The IP address that Amazon SES used to send the email.</td>
</tr>
</tbody>
</table>

How to Use Event Publishing

The following sections contain the information you need to set up and use Amazon SES event publishing.

- Setting Up Event Publishing (p. 248)
- Working with Event Data (p. 256)
- Tutorials (p. 289)
Event Publishing Terminology

The following list defines terms related to Amazon SES event publishing.

Email sending event

Information associated with the outcome of an email you submit to Amazon SES. There are eight types of email sending events:

- **Send** – Amazon SES successfully received your email and will attempt to deliver it.
- **Reject** – Amazon SES initially accepted the email, but later rejected it because the email contained a virus.
- **Bounce** – The recipient’s mail server permanently rejected the email. This event corresponds to hard bounces. Soft bounces are only included when Amazon SES fails to deliver the email after retrying for a period of time.
- **Complaint** – The message was received, but the recipient marked the email as spam.
- **Delivery** – Amazon SES successfully delivered the email to the recipient’s mail server.
- **Open** – The recipient received the message and opened it in his or her email client.
- **Click** – The recipient clicked one or more links contained in the email.
- **Rendering Failure** – The email was not sent because of a template rendering issue. This event type only occurs when you send templated email using the `SendTemplatedEmail` or `SendBulkTemplatedEmail` API operations. This event type can occur when template data is missing, or when there is a mismatch between template parameters and data.

Configuration set

An Amazon SES construct that encapsulates where you want to publish email sending events, and what email sending events you want to publish. When you send an email that you want to use with event publishing, you specify the configuration set to associate with the email.

Event destination

An Amazon SES construct that represents an AWS service to which you publish Amazon SES email sending events. Each event destination that you set up belongs to one, and only one, configuration set.

Message tag

A name/value pair that you use to categorize an email for the purpose of event publishing. Examples are `campaign/book` and `campaign/clothing`. When you send an email, you either specify the message tag as a parameter to the API call or as an Amazon SES-specific email header.

Auto-tag

Message tags that are automatically included in event publishing reports. There is an auto-tag for the configuration set name, the domain of the “From” address, the caller’s outgoing IP address, the Amazon SES outgoing IP address, and the IAM identity of the caller.

Setting Up Amazon SES Event Publishing

This section describes what you need to do to configure Amazon SES to publish your email sending events to Amazon CloudWatch or Amazon Kinesis Firehose.

You first create a configuration set using the Amazon SES console or API. After you create a configuration set, you add one or more event destinations (CloudWatch or Kinesis Firehose) to the configuration set, and configure parameters unique to the event destination. Then, each time you send an email, you
include the configuration set name and email characteristics, called message tags, as parameters to the API, or Amazon SES-specific headers in the email.

These steps are explained in the following topics.

1. Step 1: Create a Configuration Set (p. 249)
2. Add an Event Destination (p. 249)
3. Step 3: Send Email (p. 255)

**Step 1: Create a Configuration Set Using Amazon SES**

Configuration sets enable you to publish email sending events (bounces, complaints, deliveries, sent emails, and rejected emails) to Amazon CloudWatch or Amazon Kinesis Firehose.

You can use the Amazon SES console or the CreateConfigurationSet API to create a configuration set.

**To create a configuration set (console)**

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose **Configuration Sets**.
3. In the content pane, choose **Create Configuration Set**.
4. Type a name for the configuration set, and then choose **Create Configuration Set**.
5. Choose **Close**.

For information about how to use the CreateConfigurationSet API to create a configuration set, see the Amazon Simple Email Service API Reference.

**Step 2: Add an Event Destination Using Amazon SES**

Event destinations represent AWS services to which you publish email sending events such as bounces, complaints, deliveries, sent emails, and rejected emails. Each event destination that you set up belongs to one, and only one, configuration set. When you set up an event destination with Amazon SES, you choose the AWS service destination, and you specify parameters associated with that destination.

There are three event destinations: Amazon CloudWatch, Amazon Kinesis Firehose, and Amazon Simple Notification Service (Amazon SNS). The event destination that you choose depends on the level of detail you want about the events, and the way in which you want to receive the event information. If you simply want a running total of each type of event (for example, so that you can set an alarm when the total gets too high), use CloudWatch. If you want detailed event records that you can output to another service such as Amazon Elasticsearch Service or Amazon Redshift for analysis, choose Kinesis Firehose. If you want to receive notifications when certain events occur, choose Amazon SNS.

**This section contains the following topics**

- Set Up a CloudWatch Event Destination for Amazon SES Event Publishing (p. 249)
- Set Up a Kinesis Firehose Event Destination for Amazon SES Event Publishing (p. 252)
- Set Up an Amazon SNS Event Destination for Amazon SES Event Publishing (p. 254)

**Set Up a CloudWatch Event Destination for Amazon SES Event Publishing**

You can use Amazon CloudWatch event destinations to publish Amazon SES email sending events to CloudWatch. Because a CloudWatch event destination exists within a configuration set only, you must first create a configuration set (p. 249) and then add the event destination to the configuration set.
When you add a CloudWatch event destination to a configuration set, you must choose one or more CloudWatch dimensions that correspond to the message tags you use when you send your emails. Like message tags, a CloudWatch dimension is a name/value pair that helps you uniquely identify a metric.

For example, you might have a message tag and a dimension called campaign that you use to identify your email campaign. When you publish your email sending events to CloudWatch, choosing your message tags and dimensions is important because these choices affect your CloudWatch billing and determine how you can filter your email sending event data in CloudWatch.

This section provides information to help you choose your dimensions, and then shows how to add a CloudWatch event destination to a configuration set.

Topics in this section
- Adding a CloudWatch Event Destination (p. 250)
- Choosing CloudWatch Dimensions (p. 251)

Adding a CloudWatch Event Destination

The procedure in this section shows how to add a CloudWatch event destination to a configuration set.

You can also use the UpdateConfigurationSetEventDestination API operation to create event destinations. For more information about using the API, see the Amazon Simple Email Service API Reference.

To add a CloudWatch event destination to a configuration set (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation pane, choose Configuration Sets.
3. In the list of configuration sets, choose the configuration set for which you want to create a CloudWatch event destination. If the list is empty, you must first create a configuration set (p. 249).
4. On the Event Destinations tab, for Add Destination, choose Select a destination type, and then choose CloudWatch.
5. On the CloudWatch Destination dialog box, select Enabled.
6. For Name, type a name for the event destination.
7. For Event types, select the event types you want to publish to the event destination. The following event types are available:
   - **Send** – The call to Amazon SES was successful and Amazon SES will attempt to deliver the email.
   - **Reject** – Amazon SES initially accepted the email, but later rejected it because the email contained a virus.
   - **Bounce** – The recipient's mail server permanently rejected the email. This event corresponds to hard bounces. Soft bounces are only included when Amazon SES fails to deliver the email after retrying for a period of time.
   - **Complaint** – The recipient marked the email as spam.
   - **Delivery** – Amazon SES successfully delivered the email to the recipient's mail server.
   - **Open** – The recipient received the message and opened it in his or her email client.
   - **Click** – The recipient clicked one or more links contained in the email.
   - **Rendering Failure** – The email was not sent because of a template rendering issue. This event type only occurs when you send email using the SendTemplatedEmail or SendBulkTemplatedEmail API operations. This event type can occur when template data is missing, or when there is a mismatch between template parameters and data.
8. For **Value Source**, choose the location from which Amazon SES obtains the dimensions. The following value sources are available:

- **Message Tag** – Amazon SES retrieves the dimension name and value from a message tag that you specified using an `X-SES-MESSAGE-TAGS` header or a parameter to the API. For more information about adding message tags to your email, see the section called “Step 3: Send Email” (p. 255).

- **Email Header** – Amazon SES retrieves the dimension name and value from a header in the email.

  **Note**
  
  If you want to use an Amazon SES auto-tag, you must explicitly type the name of the auto-tag (such as `ses:configuration-set`) as the **Dimension Name**. For a list of auto-tags, see **How Event Publishing Works** (p. 247).

  You can't use any of the following email headers as the **Dimension Name**:

  - `Received`, `To`, `From`, `DKIM-Signature`, `CC`, `message-id`, or `Return-Path`.

- **Link Tag** – Amazon SES retrieves the dimension name and value from a tag that you specified in a link. For more information about adding tags to links, see **Can I tag links with unique identifiers?** (p. 453).

9. For **Dimension Name**, type a string to represent this email characteristic in CloudWatch. For **Default Value**, type a default value for the dimension.

  **Note**
  
  Dimension names and values can only contain the letters A through Z, the numbers 0 through 9, underscores (_), at signs (@), hyphens (-), and periods (.). Spaces, accented characters, non-Latin characters, and other special characters are not allowed.

10. Choose **Add Dimension** to add dimensions, as needed.

11. Choose **Save**.

### Choosing CloudWatch Dimensions

When you choose names and values to use as CloudWatch dimensions, consider the following factors:

- **Auto-tags** – Amazon SES automatically applies several tags (known as auto-tags) to the emails you send. To set up a CloudWatch dimension for an auto-tag, you must type the auto-tag name as the dimension name (such as `ses:configuration-set`) when you add the CloudWatch event destination to your configuration set. For a list of auto-tags and their descriptions, see **How Event Publishing Works** (p. 247).

  **Note**
  
  You can't use any of the following email headers as the **Dimension Name**:

  - `Received`, `To`, `From`, `DKIM-Signature`, `CC`, `message-id`, or `Return-Path`.

- **Price per metric** – You can view basic Amazon SES metrics in CloudWatch for free. However, when you collect metrics using event publishing, you create custom metrics in CloudWatch. Each unique combination of event type, dimension name, and dimension value creates a different custom metric in CloudWatch. When you use CloudWatch, you are charged for each custom metric you create. For this reason, you might want to avoid choosing dimensions that can take many different values. For example, unless you are very interested in tracking your email sending events by "From" domain, you might not want to define a dimension for the Amazon SES auto-tag `ses:from-domain` because it can take many different values. For more information, see **CloudWatch Pricing**.

- **Metric filtering** – If a metric has multiple dimensions, you cannot access the metric in CloudWatch based on each dimension separately. For that reason, think carefully before you add more than one dimension to a single CloudWatch event destination. For example, if you want metrics per campaign and per a combination of campaign and genre, you need to add two event destinations: one with only campaign as a dimension, and one with campaign and genre as dimensions.

- **Dimension value source** – As an alternative to specifying your dimension values using Amazon SES-specific headers or a parameter to the API, you can also choose for Amazon SES to take the dimension values from your own MIME message headers. You might use this option if you are already using
custom headers and you do not want to change your emails or your calls to the email sending API to collect metrics based on your header values. If you use your own MIME message headers for Amazon SES event publishing, the header names and values that you use for Amazon SES event publishing may only include the letters A through Z, the numbers 0 through 9, underscores (_), at signs (@), hyphens (-), and periods (.). If you specify a name or value that contains other characters, the email sending call will still succeed, but the event metrics will not be sent to Amazon CloudWatch.

For more information about CloudWatch concepts, see Amazon CloudWatch Concepts in the Amazon CloudWatch User Guide.

Set Up a Kinesis Firehose Event Destination for Amazon SES Event Publishing

An Amazon Kinesis Firehose event destination represents an entity that publishes specific Amazon SES email sending events to Kinesis Firehose. Because a Kinesis Firehose event destination exists within a configuration set only, you must first create a configuration set (p. 249) and then add the event destination to the configuration set.

You can use the Amazon SES console or the UpdateConfigurationSetEventDestination API to add a Kinesis Firehose event destination.

To add a Kinesis Firehose event destination to a configuration set (console)

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose Configuration Sets.
3. Choose a configuration set from the configuration set list. If the list is empty, you must first create a configuration set (p. 249).
4. For Add Destination, choose Select a destination type, and then choose Kinesis Firehose.
5. For Name, type a name for the event destination.
6. For Event types, select at least one event type to publish to the event destination:
   - Send – The call to Amazon SES was successful and Amazon SES will attempt to deliver the email.
   - Reject – Amazon SES initially accepted the email, but later rejected it because the email contained a virus.
   - Bounce – The recipient's mail server permanently rejected the email. This event corresponds to hard bounces. Soft bounces are only included when Amazon SES fails to deliver the email after retrying for a period of time.
   - Complaint – The recipient marked the email as spam.
   - Delivery – Amazon SES successfully delivered the email to the recipient's mail server.
   - Open – The recipient received the message and opened it in his or her email client.
   - Click – The recipient clicked one or more links contained in the email.
   - Rendering Failure – The email was not sent because of a template rendering issue. This event type only occurs when you send email using the SendTemplatedEmail or SendBulkTemplatedEmail API operations. This event type can occur when template data is missing, or when there is a mismatch between template parameters and data.
7. Select Enabled.
8. For Stream, choose an existing Kinesis Firehose delivery stream, or choose Create new stream to create a new one using the Kinesis Firehose console.
   For information about creating a stream using the Kinesis Firehose console, see Creating an Amazon Kinesis Firehose Delivery Stream in the Amazon Kinesis Data Firehose Developer Guide.
9. For IAM role, choose an IAM role for which Amazon SES has permission to publish to Kinesis Firehose on your behalf. You can choose an existing role, have Amazon SES create a role for you, or create your own role.
If you choose an existing role or create your own role, you must manually modify the role’s policies to give the role permission to access the Kinesis Firehose delivery stream, and to give Amazon SES permission to assume the role. For example policies, see Giving Amazon SES Permission to Publish to Your Kinesis Firehose Delivery Stream (p. 253).

10. Choose Save.

For information about how to use the UpdateConfigurationSetEventDestination API to add a Kinesis Firehose event destination, see the Amazon Simple Email Service API Reference.

Giving Amazon SES Permission to Publish to Your Kinesis Firehose Delivery Stream

To enable Amazon SES to publish records to your Kinesis Firehose delivery stream, you must use an AWS Identity and Access Management (IAM) role and attach or modify the role’s permissions policy and trust policy. The permissions policy enables the role to publish records to your Kinesis Firehose delivery stream, and the trust policy enables Amazon SES to assume the role.

This section provides examples of both policies. For information about attaching policies to IAM roles, see Modifying a Role in the IAM User Guide.

Permissions Policy

The following permissions policy enables the role to publish data records to your Kinesis Firehose delivery stream.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Action": ["firehose:PutRecordBatch"],
        }
    ]
}
```

Trust Policy

The following trust policy enables Amazon SES to assume the role.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {"Service": "ses.amazonaws.com"},
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "ACCOUNT-ID"
                }
            }
        }
    ]
}
```
Set Up an Amazon SNS Event Destination for Amazon SES Event Publishing

An Amazon Simple Notification Service event destination notifies you about specific email sending events using Amazon SNS. Because an Amazon SNS event destination exists within a configuration set only, you must first create a configuration set (p. 249) and then add the event destination to the configuration set.

You can use the Amazon SES console or the UpdateConfigurationSetEventDestination API operation to add an Amazon SNS event destination.

Note
It is also possible to receive notifications through Amazon SNS at the account level. This means that you can receive Amazon SNS notifications every time a sending event occurs across your entire Amazon SES account. By using event publishing rather than account-level notifications, you can configure Amazon SES to only send notifications about specific event types, or only for emails sent using a particular configuration set. For more information about setting up account-level Amazon SNS notifications, see Monitoring Using Amazon SES Notifications (p. 226).

To add an Amazon SNS event destination to a configuration set

1. If you have not already done so, create an Amazon SNS topic and subscribe to it. For more information, see Getting Started in the Amazon Simple Notification Service Developer Guide.
2. Open the Amazon SES console at https://console.aws.amazon.com/ses/.
3. In the navigation pane, choose Configuration Sets.
4. Choose a configuration set from the configuration set list. If the list is empty, you must first create a configuration set (p. 249).
5. For Add Destination, choose Select a destination type, and then choose SNS.
6. For Name, type a name for the event destination.
7. For Event types, select at least one event type to publish to the event destination:
   - Send – The call to Amazon SES was successful and Amazon SES will attempt to deliver the email.
   - Reject – Amazon SES initially accepted the email, but later rejected it because the email contained a virus.
   - Bounce – The recipient's mail server permanently rejected the email. This event corresponds to hard bounces. Soft bounces are only included when Amazon SES fails to deliver the email after retrying for a period of time.
   - Complaint – The recipient marked the email as spam.
   - Delivery – Amazon SES successfully delivered the email to the recipient's mail server.
   - Open – The recipient received the message and opened it in his or her email client.
   - Click – The recipient clicked one or more links contained in the email.
   - Rendering Failure – The email was not sent because of a template rendering issue.
     This event type only occurs when you send email using the SendTemplatedEmail or SendBulkTemplatedEmail API operations. This event type can occur when template data is missing, or when there is a mismatch between template parameters and data.
8. Select Enabled.
9. For Topic, choose an existing Amazon SNS topic, or choose Create new topic to create a new one.

For information about creating a topic, see Create a Topic in the Amazon Simple Notification Service Developer Guide.

254
Step 3: Send Email Using Amazon SES Event Publishing

After you create a configuration set (p. 249) and add an event destination (p. 249), the last step to event publishing is to send your emails.

To publish events associated with an email, you must provide the name of the configuration set to associate with the email. Optionally, you can provide message tags to categorize the email.

You provide this information to Amazon SES as either parameters to the email sending API, Amazon SES-specific email headers, or custom headers in your MIME message. The method you choose depends on which email sending interface you use, as shown in the following table.

<table>
<thead>
<tr>
<th>Email Sending Interface</th>
<th>Ways to Publish Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>SendEmail</td>
<td>API parameters</td>
</tr>
<tr>
<td>SendRawEmail</td>
<td>API parameters, Amazon SES-specific email headers, or custom MIME headers</td>
</tr>
<tr>
<td>SMTP interface</td>
<td>Amazon SES-specific email headers</td>
</tr>
</tbody>
</table>

Important
If you specify message tags using both headers and API parameters, Amazon SES uses only the message tags provided by the API parameters. Amazon SES does not join message tags specified by API parameters and headers.

The following sections describe how to specify the configuration set and message tags using headers and using API parameters.

- Using Amazon SES API Parameters (p. 255)
- Using Amazon SES-Specific Email Headers (p. 255)
- Using Custom Email Headers (p. 256)

Additionally, this guide contains several code examples that demonstrate how to send email programmatically using Amazon SES. Each of these code examples includes a method of passing a configuration set when sending an email. For more information, see Amazon SES Code Examples (p. 379).

Using Amazon SES API Parameters
To use SendEmail or SendRawEmail with event publishing, you specify the configuration set and the message tags by passing data structures called ConfigurationSet and MessageTag to the API call.

For more information about using the Amazon SES API, see the Amazon Simple Email Service API Reference.

Using Amazon SES-Specific Email Headers
When you use SendRawEmail or the SMTP interface, you can specify the configuration set and the message tags by adding Amazon SES-specific headers to the email. Amazon SES removes the headers before sending the email. The following table shows the names of the headers to use.
Event Publishing Information  
| Configuration set       | X-SES-CONFIGURATION-SET |
| Message tags            | X-SES-MESSAGE-TAGS     |

The following example shows how the headers might look in a raw email that you submit to Amazon SES.

```
X-SES-MESSAGE-TAGS: tagName1=tagValue1, tagName2=tagValue2
X-SES-CONFIGURATION-SET: myConfigurationSet
From: sender@example.com
To: recipient@example.com
Subject: Subject
Content-Type: multipart/alternative;
boundary="----=_boundary"

body
```

Using Custom Email Headers

Although you must specify the configuration set name using the Amazon SES-specific header X-SES-CONFIGURATION-SET, you can specify the message tags by using your own MIME headers.

**Note**

Header names and values that you use for Amazon SES event publishing must be in ASCII. If you specify a non-ASCII header name or value for Amazon SES event publishing, the email sending call will still succeed, but the event metrics will not be emitted to Amazon CloudWatch.

Working with Amazon SES Event Data

After you set up event publishing (p. 248) and specify a configuration set for sending emails, you can retrieve your email sending events from the event destination that you specified when you set up the configuration set associated with the email.

This section describes how to retrieve your email sending events from Amazon CloudWatch and Amazon Kinesis Firehose, and how to interpret event data provided by Amazon SNS.

- Retrieving Amazon SES Event Data from CloudWatch (p. 256)
- Retrieving Amazon SES Event Data from Kinesis Firehose (p. 258)
- Interpreting Amazon SES Event Data from Amazon SNS (p. 273)

Retrieving Amazon SES Event Data from CloudWatch

Amazon SES can publish metrics for your email sending events to Amazon CloudWatch. When you publish event data to CloudWatch, it provides these metrics as an ordered set of time-series data. You
can use these metrics to monitor the performance of your email sending. For example, you can monitor the complaint metric and set a CloudWatch alarm to trigger when the metric exceeds a certain value.

There are two levels of granularity at which Amazon SES can publish these events to CloudWatch:

- **Across your AWS account** – These coarse metrics, which correspond to the metrics you monitor using the Amazon SES console and the `GetSendStatistics` API, are totals across your entire AWS account. Amazon SES publishes these metrics to CloudWatch automatically.
- **Fine-grained** – These metrics are categorized by email characteristics that you define using message tags. To publish these metrics to CloudWatch, you must set up event publishing (p. 248) with a CloudWatch event destination and specify a configuration set (p. 255) when you send an email. You can also specify message tags or use auto-tags (p. 247) that Amazon SES automatically provides.

This section describes the available metrics and how to view the metrics in CloudWatch.

**Available Metrics**

You can publish the following Amazon SES email sending metrics to CloudWatch:

- **Send** – Your API call to Amazon SES was successful and Amazon SES attempted to deliver the email.
- **Reject** – Amazon SES initially accepted the email, but later rejected it because the email contained a virus.
- **Bounce** – The recipient's mail server permanently rejected the email. This event corresponds to hard bounces. Soft bounces are only included when Amazon SES fails to deliver the email after retrying for a period of time.
- **Complaint** – The recipient marked the email as spam.
- **Delivery** – Amazon SES successfully delivered the email to the recipient's mail server.
- **Open** – The recipient received the message and opened it in his or her email client.
- **Click** – The recipient clicked one or more links contained in the email.
- **Rendering Failure** – The email was not sent because of a template rendering issue. This event type only occurs when you send templated email using the `SendTemplatedEmail` or `SendBulkTemplatedEmail` API operations. This event type can occur when template data is missing, or when there is a mismatch between template parameters and data.

**Available Dimensions**

CloudWatch uses the dimension names that you specify when you add a CloudWatch event destination to a configuration set in Amazon SES. For more information, see Set Up a CloudWatch Event Destination for Amazon SES Event Publishing (p. 249).

**Viewing Amazon SES Metrics in the CloudWatch Console**

The following procedure describes how to view your Amazon SES event publishing metrics using the CloudWatch console.

**To view metrics using the CloudWatch console**

1. Sign in to the AWS Management Console and open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region where your AWS resources reside. For more information, see Regions and Endpoints.
3. In the navigation pane, choose **Metrics**.
4. In the **All metrics** pane, expand **AWS Namespaces**, and then choose **SES**.
5. To view metrics across your entire AWS account, which Amazon SES publishes automatically, choose **Account Sending Metrics**. To view fine-grained event publishing metrics (p. 247), choose the combination of dimensions that you specified when you set up your CloudWatch event destination (p. 249).

6. Choose the metric you want to view.

   The graph will display the metric over time.

**To view metrics using the AWS CLI**

- At a command prompt, use the following command:

```
aws cloudwatch list-metrics --namespace "AWS/SES"
```

---

**Retrieving Amazon SES Event Data from Kinesis Firehose**

Amazon SES publishes email sending events to Kinesis Firehose as JSON records. Kinesis Firehose then publishes the records to the AWS service destination that you chose when you set up the delivery stream in Kinesis Firehose. For information about setting up Kinesis Firehose delivery streams, see Creating an Amazon Kinesis Firehose Delivery Stream in the *Amazon Kinesis Data Firehose Developer Guide*.

For examples of how you can use Kinesis Firehose to publish your email sending events to Amazon Redshift and Amazon Elasticsearch Service, see *Tutorials* (p. 289).

For a description of the record contents and for example records, see the following sections.

- Event Record Contents (p. 258)
- Event Record Examples (p. 265)

---

**Contents of Amazon SES Event Data Published to Kinesis Firehose**

Amazon SES publishes email sending event records to Amazon Kinesis Firehose in JSON format. When publishing events to Kinesis Firehose, Amazon SES follows each JSON record with a newline character.

The top-level JSON object contains an eventType string, a mail object, and either a bounce, complaint, delivery, send, reject, open or click object, depending on the type of event.

See the following sections for descriptions of the different types of objects:

- Top-level JSON object (p. 258)
- mail object (p. 259)
- bounce object (p. 260)
- complaint object (p. 263)
- delivery object (p. 264)
- send object (p. 264)
- reject object (p. 264)
- open object (p. 265)
- click object (p. 265)

**Top-Level JSON Object**

The top-level JSON object in an email sending event record contains the following fields.
### Field Name | Description
--- | ---
eventType | A string that describes the type of event. Possible values: Delivery, Send, Reject, Open, Click, Bounce, Complaint, or Rendering Failure.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mail</td>
<td>A JSON object that contains information about the email that produced the event.</td>
</tr>
<tr>
<td>bounce</td>
<td>This field is only present if eventType is Bounce. It contains information about the bounce.</td>
</tr>
<tr>
<td>complaint</td>
<td>This field is only present if eventType is Complaint. It contains information about the complaint.</td>
</tr>
<tr>
<td>delivery</td>
<td>This field is only present if eventType is Delivery. It contains information about the delivery.</td>
</tr>
<tr>
<td>send</td>
<td>This field is only present if eventType is Send.</td>
</tr>
<tr>
<td>reject</td>
<td>This field is only present if eventType is Reject. It contains information about the rejection.</td>
</tr>
<tr>
<td>open</td>
<td>This field is only present if eventType is Open. It contains information about the open event.</td>
</tr>
<tr>
<td>click</td>
<td>This field is only present if eventType is Click. It contains information about the click event.</td>
</tr>
<tr>
<td>failure</td>
<td>This field is only present if eventType is Rendering Failure. It contains information about the rendering failure event.</td>
</tr>
</tbody>
</table>

### Mail Object

Each email sending event record contains information about the original email in the `mail` object. The JSON object that contains information about a `mail` object has the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp</td>
<td>The date and time, in ISO8601 format (<code>YYYY-MM-DDThh:mm:ss.sZ</code>), when the message was sent.</td>
</tr>
<tr>
<td>messageId</td>
<td>A unique ID that Amazon SES assigned to the message. Amazon SES returned this value to you when you sent the message. <strong>Note</strong> This message ID was assigned by Amazon SES. You can find the message ID of the original email in the <code>headers</code> and <code>commonHeaders</code> fields of the <code>mail</code> object.</td>
</tr>
<tr>
<td>source</td>
<td>The email address that the message was sent from (the envelope MAIL FROM address).</td>
</tr>
</tbody>
</table>
### Field Name | Description
--- | ---
sourceArn | The Amazon Resource Name (ARN) of the identity that was used to send the email. In the case of sending authorization, the sourceArn is the ARN of the identity that the identity owner authorized the delegate sender to use to send the email. For more information about sending authorization, see Using Sending Authorization (p. 132).
sendingAccountId | The AWS account ID of the account that was used to send the email. In the case of sending authorization, the sendingAccountId is the delegate sender's account ID.
destination | A list of email addresses that were recipients of the original mail.
headersTruncated | A string that specifies whether the headers are truncated in the notification, which occurs if the headers are larger than 10 KB. Possible values are true and false.
headers | A list of the email's original headers. Each header in the list has a name field and a value field.
  **Note**
  Any message ID within the headers field is from the original message that you passed to Amazon SES. The message ID that Amazon SES subsequently assigned to the message is in the messageId field of the mail object.
commonHeaders | A list of the email's original, commonly used headers. Each header in the list has a name field and a value field.
  **Note**
  Any message ID within the commonHeaders field is from the original message that you passed to Amazon SES. The message ID that Amazon SES subsequently assigned to the message is in the messageId field of the mail object.

### Bounce Object
The JSON object that contains information about a Bounce event will always have the following fields.

| Field Name     | Description |
---              | ---         |
bounceType       | The type of bounce, as determined by Amazon SES. |
bounceSubType    | The subtype of the bounce, as determined by Amazon SES. |
Field Name | Description
--- | ---
`bouncedRecipients` | A list that contains information about the recipients of the original mail that bounced.
`timestamp` | The date and time, in ISO8601 format (YYYY-MM-DDThh:mm:ss.sZ), when the ISP sent the bounce notification.
`feedbackId` | A unique ID for the bounce.
`reportingMTA` | The value of the Reporting-MTA field from the DSN. This is the value of the Message Transfer Authority (MTA) that attempted to perform the delivery, relay, or gateway operation described in the DSN.

**Note**
This field only appears if a delivery status notification (DSN) was attached to the bounce.

Bounced Recipients

A bounce event may pertain to a single recipient or to multiple recipients. The `bouncedRecipients` field holds a list of objects—one object per recipient to whom the bounce event pertains—and will always contain the following field.

Field Name | Description
--- | ---
`emailAddress` | The email address of the recipient. If a DSN is available, this is the value of the Final-Recipient field from the DSN.

Optionally, if a DSN is attached to the bounce, the following fields may also be present.

Field Name | Description
--- | ---
`action` | The value of the Action field from the DSN. This indicates the action performed by the reporting MTA as a result of its attempt to deliver the message to this recipient.
`status` | The value of the Status field from the DSN. This is the per-recipient transport-independent status code that indicates the delivery status of the message.
`diagnosticCode` | The status code issued by the reporting MTA. This is the value of the Diagnostic-Code field from the DSN. This field may be absent in the DSN (and therefore also absent in the JSON).
Bounce Types

Each bounce event will be of one of the types shown in the following table.

The event publishing system only publishes hard bounces and soft bounces that will no longer be retried by Amazon SES. When you receive bounces marked Permanent, you should remove the corresponding email addresses from your mailing list; you will not be able to send to them in the future. Transient bounces are sent to you when a message has soft bounced several times, and Amazon SES has stopped trying to re-deliver it. You may be able to successfully resend to an address that initially resulted in a Transient bounce in the future.

<table>
<thead>
<tr>
<th>bounceType</th>
<th>bounceSubType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undetermined</td>
<td>Undetermined</td>
<td>Amazon SES was unable to determine a specific bounce reason.</td>
</tr>
<tr>
<td>Permanent</td>
<td>General</td>
<td>Amazon SES received a general hard bounce. If you receive this type of bounce, you should remove the recipient's email address from your mailing list.</td>
</tr>
<tr>
<td>Permanent</td>
<td>NoEmail</td>
<td>Amazon SES received a permanent hard bounce because the target email address does not exist. If you receive this type of bounce, you should remove the recipient's email address from your mailing list.</td>
</tr>
<tr>
<td>Permanent</td>
<td>Suppressed</td>
<td>Amazon SES has suppressed sending to this address because it has a recent history of bouncing as an invalid address. For information about how to remove an address from the suppression list, see Removing an Email Address from the Amazon SES Suppression List (p. 428).</td>
</tr>
<tr>
<td>Transient</td>
<td>General</td>
<td>Amazon SES received a general bounce. You may be able to successfully send to this recipient in the future.</td>
</tr>
<tr>
<td>Transient</td>
<td>MailboxFull</td>
<td>Amazon SES received a mailbox full bounce. You may be able to successfully send to this recipient in the future.</td>
</tr>
<tr>
<td>Transient</td>
<td>MessageTooLarge</td>
<td>Amazon SES received a message too large bounce. You may be able to successfully send to this recipient if you reduce the size of the message.</td>
</tr>
<tr>
<td>Transient</td>
<td>ContentRejected</td>
<td>Amazon SES received a content rejected bounce. You may be able to successfully send to this recipient if you change the content of the message.</td>
</tr>
<tr>
<td>Transient</td>
<td>AttachmentRejected</td>
<td>Amazon SES received an attachment rejected bounce. You may be able to successfully send to this recipient if you remove or change the attachment.</td>
</tr>
</tbody>
</table>
Complaint Object

The JSON object that contains information about a Complaint event has the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>complainedRecipients</td>
<td>A list that contains information about recipients that may have submitted the complaint.</td>
</tr>
<tr>
<td>timestamp</td>
<td>The date and time, in ISO8601 format (YYYY-MM-DDTh:mm:ss.sZ), when the ISP sent the complaint notification.</td>
</tr>
<tr>
<td>feedbackId</td>
<td>A unique ID for the complaint.</td>
</tr>
</tbody>
</table>

Further, if a feedback report is attached to the complaint, the following fields may be present.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userAgent</td>
<td>The value of the User-Agent field from the feedback report. This indicates the name and version of the system that generated the report.</td>
</tr>
<tr>
<td>complaintFeedbackType</td>
<td>The value of the Feedback-Type field from the feedback report received from the ISP. This contains the type of feedback.</td>
</tr>
<tr>
<td>arrivalDate</td>
<td>The value of the Arrival-Date or Received-Date field from the feedback report in ISO8601 format (YYYY-MM-DDTh:mm:ss.sZ). This field may be absent in the report (and therefore also absent in the JSON).</td>
</tr>
</tbody>
</table>

Complained Recipients

The complainedRecipients field contains a list of recipients that may have submitted the complaint.

**Important**
Since most ISPs redact the email address of the recipient who submitted the complaint from their complaint notification, this list contains information about recipients who might have sent the complaint, based on the recipients of the original message and the ISP from which we received the complaint. Amazon SES performs a lookup against the original message to determine this recipient list.

JSON objects in this list contain the following field.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>emailAddress</td>
<td>The email address of the recipient.</td>
</tr>
</tbody>
</table>

Complaint Types

You may see the following complaint types in the complaintFeedbackType field as assigned by the reporting ISP, according to the Internet Assigned Numbers Authority website:
### Field Name | Description
---|---
abuse | Indicates unsolicited email or some other kind of email abuse.
auth-failure | Email authentication failure report.
fraud | Indicates some kind of fraud or phishing activity.
not-spam | Indicates that the entity providing the report does not consider the message to be spam. This may be used to correct a message that was incorrectly tagged or categorized as spam.
other | Indicates any other feedback that does not fit into other registered types.
virus | Reports that a virus is found in the originating message.

delivery Object

The JSON object that contains information about a Delivery event will always have the following fields.

| Field Name          | Description                                                                                                                                 |
---|---|
timestamp | The date and time when Amazon SES delivered the email to the recipient's mail server, in ISO8601 format (YYYY-MM-DDThh:mm:ss.sZ).
processingTimeMillis | The time in milliseconds between when Amazon SES accepted the request from the sender to when Amazon SES passed the message to the recipient's mail server.
recipients | A list of intended recipients that the delivery event applies to.
smtphResponse | The SMTP response message of the remote ISP that accepted the email from Amazon SES. This message will vary by email, by receiving mail server, and by receiving ISP.
reportingMTA | The host name of the Amazon SES mail server that sent the mail.

send Object

The JSON object that contains information about a send event is always empty.

Reject Object

The JSON object that contains information about a Reject event will always have the following fields.
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reason</td>
<td>The reason the email was rejected. The only possible value is <code>Bad content</code>, which means that Amazon SES detected that the email contained a virus.</td>
</tr>
</tbody>
</table>

**Open Object**

The JSON object that contains information about a open event will always contain the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipAddress</td>
<td>The recipient's IP address.</td>
</tr>
<tr>
<td>timestamp</td>
<td>The date and time when the open event occurred in ISO8601 format (<code>YYYY-MM-DDThh:mm:ss.sZ</code>).</td>
</tr>
<tr>
<td>userAgent</td>
<td>The user agent of the device or email client that the recipient used to open the email.</td>
</tr>
</tbody>
</table>

**Click Object**

The JSON object that contains information about a click event will always contain the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipAddress</td>
<td>The recipient's IP address.</td>
</tr>
<tr>
<td>timestamp</td>
<td>The date and time when the click event occurred in ISO8601 format (<code>YYYY-MM-DDThh:mm:ss.sZ</code>).</td>
</tr>
<tr>
<td>userAgent</td>
<td>The user agent of the client that the recipient used to click a link in the email.</td>
</tr>
<tr>
<td>link</td>
<td>The URL of the link that the recipient clicked.</td>
</tr>
<tr>
<td>linkTags</td>
<td>A list of tags that were added to the link using the <code>ses:tags</code> attribute. For more information about adding tags to links in your emails, see Q5. Can I tag links with unique identifiers? (p. 453) in the Amazon SES Email Sending Metrics FAQs (p. 450).</td>
</tr>
</tbody>
</table>

**Examples of Amazon SES Event Data Published to Kinesis Firehose**

This section provides examples of each type of email sending event record that Amazon SES publishes to Kinesis Firehose.

The event types are as follows:

- Bounce Record (p. 266)
- Complaint Record (p. 266)
- Delivery Record (p. 267)
Bounce Record

The following is an example of a bounce event record that Amazon SES publishes to Kinesis Firehose.

```
{
  "notificationType":"Bounce",
  "bounce":{
    "bounceType":"Permanent",
    "bounceSubType":"General",
    "bouncedRecipients": [
      {
        "emailAddress":"recipient@example.com",
        "action":"failed",
        "status":"5.1.1",
        "diagnosticCode":"smtp; 550 5.1.1 user unknown"
      }
    ],
    "timestamp":"2018-01-22T18:20:34.063Z",
    "feedbackId":"01000157c44f053b-61b59c11-9236-11e6-8f96-7be8a4ae61bb-000000",
    "remoteMtaiP":"0.0.0.0",
    "reportingMTA":"dsn; a27-57.smtp-out.us-west-2.amazonses.com"
  },
  "mail":{
    "timestamp":"2018-01-22T18:20:33.000Z",
    "source":"sender@example.com",
    "sourceArn":"arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sourceIp":"192.0.2.0",
    "sendingAccountId":"123456789012",
    "messageId":"EXAMPLE7c191be45-e9aedd9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "destination": [
      "recipient@example.com"
    ]
  }
}
```

Complaint Record

The following is an example of a complaint event record that Amazon SES publishes to Kinesis Firehose.

```
{
  "notificationType":"Complaint",
  "complaint":{
    "complainedRecipients": [
      {
        "emailAddress":"recipient@example.com"
      }
    ],
    "timestamp":"2018-01-22T18:20:34.000Z",
    "feedbackId":"01000157c44f053b-61b59c11-9236-11e6-8f96-7be8a4ae61bb-000000",
    "userAgent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.90 Safari/537.36",
    "complaintFeedbackType":"abuse"
  },
  "mail":{
    "timestamp":"2018-01-22T18:20:36.062Z",
    "sourceArn":"arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sourceIp":"192.0.2.0",
    "sendingAccountId":"123456789012",
    "messageId":"EXAMPLE7c191be45-e9aedd9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "destination": [
      "recipient@example.com"
    ]
  }
}
```
Delivery Record

The following is an example of a delivery event record that Amazon SES publishes to Kinesis Firehose.

```json
{  "eventType": "Delivery",  "mail": {    "timestamp": "2016-10-19T23:20:52.240Z",    "source": "sender@example.com",    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",    "sendingAccountId": "123456789012",    "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",    "destination": [      "recipient@example.com"    ],    "headersTruncated": false,    "headers": [      { "name": "From", "value": "sender@example.com" },      { "name": "To", "value": "recipient@example.com" },      { "name": "Subject", "value": "Email Subject" },      { "name": "MIME-Version", "value": "1.0" },      { "name": "Content-Type", "value": "text/html; charset=UTF-8" },      { "name": "Content-Transfer-Encoding", "value": "7bit" }    ],    "commonHeaders": {      "from": [        "sender@example.com"      ],      "to": [        "recipient@example.com"      ],      "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",      "subject": "Email Subject"    },    "tags": {  
```

"ses:configuration-set": [  "my-configuration-set" ],  
"ses:source-ip": [  "192.0.2.0" ],  
"ses:from-domain": [  "example.com" ],  
"ses:caller-identity": [  "ses_user" ],  
"ses:outgoing-ip": [  "192.0.2.0" ],  
"myCustomTag1": [  "myCustomTagValue1" ],  
"myCustomTag2": [  "myCustomTagValue2" ]
},
"delivery": {
  "timestamp": "2016-10-19T23:21:04.133Z",
  "processingTimeMillis": 11893,
  "recipients": [  "recipient@example.com"
  ],
  "smtpResponse": "250 2.6.0 Message received",
  "reportingMTA": "mta.example.com"
}
}

Send Email Record

The following is an example of a send event record that Amazon SES publishes to Kinesis Firehose.

{  "eventType": "Send",
  "mail": {
    "timestamp": "2016-10-14T05:02:16.645Z",
    "source": "sender@example.com",
    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sendingAccountId": "123456789012",
    "messageId": "EXAMPLE7c191be45-e9aeb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "destination": [  "recipient@example.com"
    ],
    "headersTruncated": false,
    "headers": [  
      {  "name": "From",
        "value": "sender@example.com"
      },
      {  "name": "To",
        "value": "recipient@example.com"
      },
      {  "name": "Subject",
        "value": "Email Subject"
      }
    ]
  }
}
Reject Event Record

The following is an example of a reject event record that Amazon SES publishes to Kinesis Firehose.

```json
{
    "eventType": "Reject",
    "mail": {
        "timestamp": "2016-10-14T17:38:15.211Z",
        "source": "sender@example.com",
        "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
        "sendingAccountId": "123456789012",
        "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
        "destination": [
            "sender@example.com"
        ],
        "headersTruncated": false,
        "headers": [
            {
                "name": "From",
                "value": "sender@example.com"
            }
        ]
    }
}
```
Open Event Record

The following is an example of an open event record that Amazon SES publishes to Kinesis Firehose.

```
{
    "eventType": "Open",
    "value": "sender@example.com",
    "name": "To",
    "value": "sender@example.com",
    "name": "Subject",
    "value": "Email Subject",
    "name": "MIME-Version",
    "value": "1.0",
    "name": "Content-Type",
    "value": "multipart/mixed; boundary="qMm9M+Fa2AknHoGS"
    "name": "X-SES-MESSAGE-TAGS",
    "value": "myCustomTag1=myCustomTagValue1, myCustomTag2=myCustomTagValue2"
    "commonHeaders": {
        "from": ["sender@example.com"],
        "to": ["sender@example.com"],
        "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
        "subject": "Email Subject"
    },
    "tags": {
        "ses:configuration-set": ["my-configuration-set"],
        "ses:source-ip": ["192.0.2.0"],
        "ses:from-domain": ["example.com"],
        "ses:caller-identity": ["ses_user"],
        "myCustomTag1": ["myCustomTagValue1"],
        "myCustomTag2": ["myCustomTagValue2"
    }
}
```
"mail": {
    "commonHeaders": {
        "from": [
            "sender@example.com"
        ],
        "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
        "subject": "Email Subject",
        "to": [
            "recipient@example.com"
        ],
        "headers": [
            {
                "name": "X-SES-CONFIGURATION-SET",
                "value": "ConfigSet"
            },
            {
                "name": "From",
                "value": "sender@example.com"
            },
            {
                "name": "To",
                "value": "recipient@example.com"
            },
            {
                "name": "Subject",
                "value": "Email Subject"
            },
            {
                "name": "MIME-Version",
                "value": "1.0"
            },
            {
                "name": "Content-Type",
                "value": "multipart/alternative; boundary="XBoundary""
            }
        ],
        "headersTruncated": false,
        "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
        "sendingAccountId": "123456789012",
        "source": "sender@example.com",
        "tags": {
            "ses:caller-identity": [
                "ses-user"
            ],
            "ses:configuration-set": [
                "ConfigSet"
            ],
            "ses:from-domain": [
                "example.com"
            ],
            "ses:source-ip": [
                "192.0.2.0"
            ],
        },
        "timestamp": "2017-08-09T21:59:49.927Z"
    },
    "open": {
        "ipAddress": "192.0.2.1",
        "timestamp": "2017-08-09T22:00:19.652Z",
        "userAgent": "Mozilla/5.0 (iPhone; CPU iPhone OS 10_3_3 like Mac OS X) AppleWebKit/603.3.8 (KHTML, like Gecko) Mobile/14G60"
Click Event Record

The following is an example of a click event record that Amazon SES publishes to Kinesis Firehose.

```json
{
  "eventType": "Click",
  "click": {
    "ipAddress": "192.0.2.1",
    "linkTags": {
      "samplekey0": ["samplevalue0"],
      "samplekey1": ["samplevalue1"]
    },
    "timestamp": "2017-08-09T23:51:25.570Z",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.90 Safari/537.36"
  },
  "mail": {
    "commonHeaders": {
      "from": ["sender@example.com"],
      "messageId": "EXAMPLE7c191be45-e9aeb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
      "subject": "Amazon SES Test",
      "to": ["recipient@example.com"]
    },
    "destination": ["recipient@example.com"],
    "headers": [
      {"name": "X-SES-CONFIGURATION-SET", "value": "ConfigSet"},
      {"name": "From", "value": "sender@example.com"},
      {"name": "To", "value": "recipient@example.com"},
      {"name": "Subject", "value": "Email Subject"},
      {"name": "MIME-Version", "value": "1.0"},
      {"name": "Content-Type", "value": "multipart/alternative; boundary="XBoundary""},
      {"name": "Message-ID", "value": "null"}
    ]
  }
}
```
Interpreting Amazon SES Event Data from Amazon SNS

Amazon SES publishes email sending events to Amazon Simple Notification Service (Amazon SNS) as JSON records. Amazon SNS then delivers notifications to the endpoints that are subscribed to the Amazon SNS topic associated with the event destination. For information about setting up topics and subscriptions in Amazon SNS, see Getting Started in the Amazon Simple Notification Service Developer Guide.

For a description of the record contents and for example records, see the following sections.

- Event Record Contents (p. 273)
- Event Record Examples (p. 281)

Contents of Amazon SES Event Data Published to Amazon SNS

Amazon SES publishes email sending event records to Amazon Simple Notification Service in JSON format.

The top-level JSON object contains an eventType string, a mail object, and either a bounce, complaint, delivery, send, reject, open, click, or failure object, depending on the type of event.

Topics in this section:
- Top-Level JSON Object (p. 274)
- Mail Object (p. 274)
- Bounce Object (p. 275)
- Complaint Object (p. 278)
- Delivery Object (p. 279)
- Send Object (p. 279)
- Reject Object (p. 279)
- Open Object (p. 280)
- Click Object (p. 280)
- Failure Object (p. 280)
Top-Level JSON Object

The top-level JSON object in an email sending event record contains the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventType</td>
<td>A string that describes the type of event. Possible values: Delivery, Send, Reject, Open, Click, Bounce, Complaint, or Rendering Failure.</td>
</tr>
<tr>
<td>mail</td>
<td>A JSON object that contains information about the email that produced the event.</td>
</tr>
<tr>
<td>bounce</td>
<td>This field is only present if eventType is Bounce. It contains information about the bounce.</td>
</tr>
<tr>
<td>complaint</td>
<td>This field is only present if eventType is Complaint. It contains information about the complaint.</td>
</tr>
<tr>
<td>delivery</td>
<td>This field is only present if eventType is Delivery. It contains information about the delivery.</td>
</tr>
<tr>
<td>send</td>
<td>This field is only present if eventType is Send.</td>
</tr>
<tr>
<td>reject</td>
<td>This field is only present if eventType is Reject. It contains information about the rejection.</td>
</tr>
<tr>
<td>open</td>
<td>This field is only present if eventType is Open. It contains information about the open event.</td>
</tr>
<tr>
<td>click</td>
<td>This field is only present if eventType is Click. It contains information about the click event.</td>
</tr>
<tr>
<td>failure</td>
<td>This field is only present if eventType is Rendering Failure. It contains information about the rendering failure event.</td>
</tr>
</tbody>
</table>

Mail Object

Each email sending event record contains information about the original email in the mail object. The JSON object that contains information about a mail object has the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp</td>
<td>The date and time, in ISO8601 format (YYYY-MM-DDThh:mm:ss.sZ), when the message was sent.</td>
</tr>
<tr>
<td>messageId</td>
<td>A unique ID that Amazon SES assigned to the message. Amazon SES returned this value to you when you sent the message. <strong>Note</strong> This message ID was assigned by Amazon SES. You can find the message ID of the original email in the headers and...</td>
</tr>
</tbody>
</table>
### Field Name | Description
--- | ---
| commonHeaders | commonHeaders fields of the mail object.
| source | The email address that the message was sent from (the envelope MAIL FROM address).
| sourceArn | The Amazon Resource Name (ARN) of the identity that was used to send the email. In the case of sending authorization, the sourceArn is the ARN of the identity that the identity owner authorized the delegate sender to use to send the email. For more information about sending authorization, see Using Sending Authorization (p. 132).
| sendingAccountId | The AWS account ID of the account that was used to send the email. In the case of sending authorization, the sendingAccountId is the delegate sender’s account ID.
| destination | A list of email addresses that were recipients of the original mail.
| headersTruncated | A string that specifies whether the headers are truncated in the notification, which occurs if the headers are larger than 10 KB. Possible values are `true` and `false`.
| headers | A list of the email’s original headers. Each header in the list has a `name` field and a `value` field.
  **Note**
  Any message ID within the `headers` field is from the original message that you passed to Amazon SES. The message ID that Amazon SES subsequently assigned to the message is in the `messageId` field of the `mail` object.
| commonHeaders | A list of the email’s original, commonly used headers. Each header in the list has a `name` field and a `value` field.
  **Note**
  Any message ID within the `commonHeaders` field is from the original message that you passed to Amazon SES. The message ID that Amazon SES subsequently assigned to the message is in the `messageId` field of the `mail` object.

**Bounce Object**

The JSON object that contains information about a Bounce event has the following fields.
### Field Name | Description
---|---
**bounceType** | The type of bounce, as determined by Amazon SES.
**bounceSubType** | The subtype of the bounce, as determined by Amazon SES.
**bouncedRecipients** | A list that contains information about the recipients of the original mail that bounced.
**timestamp** | The date and time, in ISO8601 format ("YYYY-MM-DDThh:mm:ss.sZ"), when the ISP sent the bounce notification.
**feedbackId** | A unique ID for the bounce.
**reportingMTA** | The value of the Reporting-MTA field from the DSN. This is the value of the Message Transfer Authority (MTA) that attempted to perform the delivery, relay, or gateway operation described in the DSN. 

**Note**
This field only appears if a delivery status notification (DSN) was attached to the bounce.

### Bounced Recipients
A bounce event may pertain to a single recipient or to multiple recipients. The **bouncedRecipients** field holds a list of objects—one object per recipient whose email address produced a bounce—and contains the following field.

| Field Name | Description |
---|---|
**emailAddress** | The email address of the recipient. If a DSN is available, this is the value of the Final-Recipient field from the DSN.

 Optionally, if a DSN is attached to the bounce, the following fields may also be present.

| Field Name | Description |
---|---|
**action** | The value of the Action field from the DSN. This indicates the action performed by the reporting MTA as a result of its attempt to deliver the message to this recipient. |
**status** | The value of the Status field from the DSN. This is the per-recipient transport-independent status code that indicates the delivery status of the message. |
**diagnosticCode** | The status code issued by the reporting MTA. This is the value of the Diagnostic-Code field from |
Field Name | Description
--- | ---
the DSN. This field may be absent in the DSN (and therefore also absent in the JSON).

### Bounce Types

Each bounce event is of one of the types shown in the following table.

The event publishing system only publishes hard bounces and soft bounces that are no longer retried by Amazon SES. When you receive bounces marked Permanent, you should remove the corresponding email addresses from your mailing list; you will not be able to send to them in the future. Transient bounces are sent to you when a message has soft bounced several times, and Amazon SES has stopped trying to re-deliver it. You may be able to successfully resend to an address that initially resulted in a Transient bounce in the future.

<table>
<thead>
<tr>
<th>bounceType</th>
<th>bounceSubType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undetermined</td>
<td>Undetermined</td>
<td>Amazon SES was unable to determine a specific bounce reason.</td>
</tr>
<tr>
<td>Permanent</td>
<td>General</td>
<td>Amazon SES received a general hard bounce. If you receive this type of bounce, you should remove the recipient's email address from your mailing list.</td>
</tr>
<tr>
<td>Permanent</td>
<td>NoEmail</td>
<td>Amazon SES received a permanent hard bounce because the target email address does not exist. If you receive this type of bounce, you should remove the recipient's email address from your mailing list.</td>
</tr>
<tr>
<td>Permanent</td>
<td>Suppressed</td>
<td>Amazon SES has suppressed sending to this address because it has a recent history of bouncing as an invalid address. For information about how to remove an address from the suppression list, see Removing an Email Address from the Amazon SES Suppression List (p. 428).</td>
</tr>
<tr>
<td>Transient</td>
<td>General</td>
<td>Amazon SES received a general bounce. You may be able to successfully send to this recipient in the future.</td>
</tr>
<tr>
<td>Transient</td>
<td>MailboxFull</td>
<td>Amazon SES received a mailbox full bounce. You may be able to successfully send to this recipient in the future.</td>
</tr>
<tr>
<td>Transient</td>
<td>MessageTooLarge</td>
<td>Amazon SES received a message too large bounce. You may be able to successfully send to this recipient if you reduce the size of the message.</td>
</tr>
<tr>
<td>Transient</td>
<td>ContentRejected</td>
<td>Amazon SES received a content rejected bounce. You may be able to successfully send to this recipient if you change the content of the message.</td>
</tr>
<tr>
<td>Transient</td>
<td>AttachmentRejected</td>
<td>Amazon SES received an attachment rejected bounce. You may be able to successfully send...</td>
</tr>
</tbody>
</table>
### Working with Event Data

<table>
<thead>
<tr>
<th>bounceType</th>
<th>bounceSubType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>to this recipient if you remove or change the attachment.</td>
</tr>
</tbody>
</table>

**Complaint Object**

The JSON object that contains information about a `complaint` event has the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>complainedRecipients</td>
<td>A list that contains information about recipients that may have submitted the complaint.</td>
</tr>
<tr>
<td>timestamp</td>
<td>The date and time, in ISO8601 format (<code>YYYY-MM-DDThh:mm:ss.sZ</code>), when the ISP sent the complaint notification.</td>
</tr>
<tr>
<td>feedbackId</td>
<td>A unique ID for the complaint.</td>
</tr>
</tbody>
</table>

Further, if a feedback report is attached to the complaint, the following fields may be present.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userAgent</td>
<td>The value of the <code>User-Agent</code> field from the feedback report. This indicates the name and version of the system that generated the report.</td>
</tr>
<tr>
<td>complaintFeedbackType</td>
<td>The value of the <code>Feedback-Type</code> field from the feedback report received from the ISP. This contains the type of feedback.</td>
</tr>
<tr>
<td>arrivalDate</td>
<td>The value of the <code>Arrival-Date</code> or <code>Received-Date</code> field from the feedback report in ISO8601 format (<code>YYYY-MM-DDThh:mm:ss.sZ</code>). This field may be absent in the report (and therefore also absent in the JSON).</td>
</tr>
</tbody>
</table>

**Complained Recipients**

The `complainedRecipients` field contains a list of recipients that may have submitted the complaint.

**Important**

Most ISPs redact the email addresses of recipients who submit complaints. For this reason, the `complainedRecipients` field includes a list of everyone who was sent the email whose address is on the domain that issued the complaint notification.

JSON objects in this list contain the following field.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>emailAddress</td>
<td>The email address of the recipient.</td>
</tr>
</tbody>
</table>
Complaint Types

You may see the following complaint types in the `complaintFeedbackType` field as assigned by the reporting ISP, according to the [Internet Assigned Numbers Authority website](https://www.iana.org/):  

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abuse</td>
<td>Indicates unsolicited email or some other kind of email abuse.</td>
</tr>
<tr>
<td>auth-failure</td>
<td>Email authentication failure report.</td>
</tr>
<tr>
<td>fraud</td>
<td>Indicates some kind of fraud or phishing activity.</td>
</tr>
<tr>
<td>not-spam</td>
<td>Indicates that the entity providing the report does not consider the message to be spam. This may be used to correct a message that was incorrectly tagged or categorized as spam.</td>
</tr>
<tr>
<td>other</td>
<td>Indicates any other feedback that does not fit into other registered types.</td>
</tr>
<tr>
<td>virus</td>
<td>Reports that a virus is found in the originating message.</td>
</tr>
</tbody>
</table>

Delivery Object

The JSON object that contains information about a `Delivery` event has the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp</td>
<td>The date and time when Amazon SES delivered the email to the recipient's mail server, in ISO8601 format (<code>YYYY-MM-DDThh:mm:ss.sZ</code>).</td>
</tr>
<tr>
<td>processingTimeMillis</td>
<td>The time in milliseconds between when Amazon SES accepted the request from the sender to when Amazon SES passed the message to the recipient's mail server.</td>
</tr>
<tr>
<td>recipients</td>
<td>A list of intended recipients that the delivery event applies to.</td>
</tr>
<tr>
<td>smtpResponse</td>
<td>The SMTP response message of the remote ISP that accepted the email from Amazon SES. This message will vary by email, by receiving mail server, and by receiving ISP.</td>
</tr>
<tr>
<td>reportingMTA</td>
<td>The host name of the Amazon SES mail server that sent the mail.</td>
</tr>
</tbody>
</table>

Send Object

The JSON object that contains information about a `send` event is always empty.

Reject Object

The JSON object that contains information about a `Reject` event has the following fields.
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reason</td>
<td>The reason the email was rejected. The only possible value is Bad content, which means that Amazon SES detected that the email contained a virus.</td>
</tr>
</tbody>
</table>

**Open Object**

The JSON object that contains information about a Open event has the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipAddress</td>
<td>The recipient's IP address.</td>
</tr>
<tr>
<td>timestamp</td>
<td>The date and time when the open event occurred in ISO8601 format (<strong>YYYY-MM-DDThh:mm:ss.sZ</strong>).</td>
</tr>
<tr>
<td>userAgent</td>
<td>The user agent of the device or email client that the recipient used to open the email.</td>
</tr>
</tbody>
</table>

**Click Object**

The JSON object that contains information about a Click event has the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipAddress</td>
<td>The recipient's IP address.</td>
</tr>
<tr>
<td>timestamp</td>
<td>The date and time when the click event occurred in ISO8601 format (<strong>YYYY-MM-DDThh:mm:ss.sZ</strong>).</td>
</tr>
<tr>
<td>userAgent</td>
<td>The user agent of the client that the recipient used to click a link in the email.</td>
</tr>
<tr>
<td>link</td>
<td>The URL of the link that the recipient clicked.</td>
</tr>
<tr>
<td>linkTags</td>
<td>A list of tags that were added to the link using the <em>ses:tags</em> attribute. For more information about adding tags to links in your emails, see Q5. Can I tag links with unique identifiers? (p. 453) in the Amazon SES Email Sending Metrics FAQs (p. 450).</td>
</tr>
</tbody>
</table>

**Failure Object**

The JSON object that contains information about a Rendering Failure event has the following fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>templateName</td>
<td>The name of the template used to send the email.</td>
</tr>
<tr>
<td>errorMessage</td>
<td>A message that provides more information about the rendering failure.</td>
</tr>
</tbody>
</table>
Examples of Amazon SES Event Data Published to Amazon SNS

This section provides examples of each type of email sending event record that Amazon SES publishes to Amazon SNS.

Topics in this section:
- Bounce Record (p. 281)
- Complaint Record (p. 281)
- Delivery Record (p. 282)
- Send Email Record (p. 283)
- Reject Event Record (p. 284)
- Open Event Record (p. 286)
- Click Event Record (p. 287)
- Failure Event Record (p. 288)

Bounce Record

The following is an example of a bounce event record that Amazon SES publishes to Amazon SNS.

```json
{
    "notificationType":"Bounce",
    "bounce":{
        "bounceType":"Permanent",
        "bounceSubType":"General",
        "bouncedRecipients":[
            {
                "emailAddress":"recipient@example.com",
                "action":"failed",
                "status":"5.1.1",
                "diagnosticCode":"smtp; 550 5.1.1 user unknown"
            }
        ],
        "timestamp":"2018-01-22T18:20:34.063Z",
        "feedbackId":"01000157c44f053b-61b59c11-9236-11e6-8f96-7be8a4ae61bb-000000",
        "remoteMtaIp":"0.00",
        "reportingMTA":"dsn; a27-57.smtp-out.us-west-2.amazonses.com"
    },
    "mail":{
        "timestamp":"2018-01-22T18:20:33.000Z",
        "source":"sender@example.com",
        "sourceArn":"arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
        "sourceIp":"192.0.2.0",
        "sendingAccountId":"123456789001",
        "messageId":"EXAMPLE7c191be45-e9aebd9a-02f9-4d12-a87d-dd0099a07f8a-000000",
        "destination":{
            "recipient@example.com"
        }
    }
}
```

Complaint Record

The following is an example of a complaint event record that Amazon SES publishes to Amazon SNS.

```json
{
    "notificationType":"Complaint",
    "complaint":{
        "complainedRecipients":[
        ]
    }
}
```
Delivery Record

The following is an example of a delivery event record that Amazon SES publishes to Amazon SNS.

```json
{
  "eventType": "Delivery",
  "mail": {
    "timestamp": "2016-10-19T23:20:52.240Z",
    "source": "sender@example.com",
    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sendingAccountId": "123456789012",
    "messageId": "EXAMPLE7c191be45-e9aeb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "destination": [
      "recipient@example.com"
    ],
    "headersTruncated": false,
    "headers": [
      {
        "name": "From",
        "value": "sender@example.com"
      },
      {
        "name": "To",
        "value": "recipient@example.com"
      },
      {
        "name": "Subject",
        "value": "Email Subject"
      },
      {
        "name": "MIME-Version",
        "value": "1.0"
      },
      {
        "name": "Content-Type",
        "value": "text/html; charset=UTF-8"
      },
      {
        "name": "Content-Transfer-Encoding",
        "value": "7bit"
      }
    ],
    "commonHeaders": {
      "emailAddress": "recipient@example.com"
    },
    "feedbackId": "01000157c44f053b-61b59c11-9236-11e6-8f96-7be8a4ae61bb-000000",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.90 Safari/537.36",
    "complaintFeedbackType": "abuse"
  }
}
```
Send Email Record

The following is an example of a send event record that Amazon SES publishes to Amazon SNS.

```json
{
  "eventType": "Send",
  "mail": {
    "timestamp": "2016-10-14T05:02:16.645Z",
    "source": "sender@example.com",
    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sendingAccountId": "123456789012",
    "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "destination": [
      "recipient@example.com"
    ],
    "headersTruncated": false,
    "headers": [
      {
        "name": "From",
        "value": "sender@example.com"
      }
    ]
  }
}
```
Reject Event Record

The following is an example of a reject event record that Amazon SES publishes to Amazon SNS.

```json
{
    "eventType": "Reject",
    "mail": {
        "timestamp": "2016-10-14T17:38:15.211Z",
        "commonHeaders": {
            "from": ["sender@example.com"],
            "to": ["recipient@example.com"],
            "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
            "subject": "Email Subject"
        },
        "tags": {
            "ses:configuration-set": ["my-configuration-set"],
            "ses:source-ip": ["192.0.2.0"],
            "ses:from-domain": ["example.com"],
            "ses:caller-identity": ["ses_user"],
            "myCustomTag1": ["myCustomTagValue1"],
            "myCustomTag2": ["myCustomTagValue2"],
            "X-SES-MESSAGE-TAGS": "myCustomTag1=myCustomTagValue1, myCustomTag2=myCustomTagValue2"
        }
    }
}
```
"source": "sender@example.com",
"sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
"sendingAccountArn": "123456789012",
"messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
"destination": [
  "sender@example.com"
],
"headersTruncated": false,
"headers": [
  {
    "name": "From",
    "value": "sender@example.com"
  },
  {
    "name": "To",
    "value": "sender@example.com"
  },
  {
    "name": "Subject",
    "value": "Email Subject"
  },
  {
    "name": "MIME-Version",
    "value": "1.0"
  },
  {
    "name": "Content-Type",
    "value": "multipart/mixed; boundary="qMm9M+Fa2AknHoGS"
  },
  {
    "name": "X-SES-MESSAGE-TAGS",
    "value": "myCustomTag1=myCustomTagValue1, myCustomTag2=myCustomTagValue2"
  }
],
"commonHeaders": {
  "from": ["sender@example.com"],
  "to": ["sender@example.com"],
  "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
  "subject": "Email Subject"
},
"tags": {
  "ses:configuration-set": ["my-configuration-set"],
  "ses:source-ip": ["192.0.2.0"],
  "ses:from-domain": ["example.com"],
  "ses:caller-identity": ["ses_user"],
  "myCustomTag1": ["myCustomTagValue1"],
  "myCustomTag2": ["myCustomTagValue2"
  ]
},
"reject": {
Open Event Record

The following is an example of an open event record that Amazon SES publishes to Amazon SNS.

```json
{
  "eventType": "Open",
  "mail": {
    "commonHeaders": {
      "from": [
        "sender@example.com"
      ],
      "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
      "subject": "Email Subject",
      "to": [
        "recipient@example.com"
      ],
      "destination": [
        "recipient@example.com"
      ],
      "headers": [
        {
          "name": "X-SES-CONFIGURATION-SET",
          "value": "ConfigSet"
        },
        {
          "name": "From",
          "value": "sender@example.com"
        },
        {
          "name": "To",
          "value": "recipient@example.com"
        },
        {
          "name": "Subject",
          "value": "Email Subject"
        },
        {
          "name": "MIME-Version",
          "value": "1.0"
        },
        {
          "name": "Content-Type",
          "value": "multipart/alternative; boundary="XBoundary""
        }
      ],
      "headersTruncated": false,
      "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
      "sendingAccountId": "123456789012",
      "source": "sender@example.com",
      "tags": {
        "ses:caller-identity": [
          "ses-user"
        ],
        "ses:configuration-set": [
          "ConfigSet"
        ],
        "ses:from-domain": [
          "example.com"
        ]
      }
    },
    "headersTruncated": false,
    "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "sendingAccountId": "123456789012",
    "source": "sender@example.com",
    "tags": {
      "ses:caller-identity": [
        "ses-user"
      ],
      "ses:configuration-set": [
        "ConfigSet"
      ],
      "ses:from-domain": [
        "example.com"
      ]
    }
  }
}
```
Click Event Record

The following is an example of a click event record that Amazon SES publishes to Amazon SNS.

```json
{
    "eventType": "Click",
    "click": {
        "ipAddress": "192.0.2.1",
        "linkTags": {
            "samplekey0": ["samplevalue0"],
            "samplekey1": ["samplevalue1"
        ],
        "timestamp": "2017-08-09T23:51:25.570Z",
        "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.90 Safari/537.36"
    },
    "mail": {
        "commonHeaders": {
            "from": ["sender@example.com"
        ],
        "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
        "subject": "Amazon SES Test",
        "to": ["recipient@example.com"
        ],
        "destination": ["recipient@example.com"
        ],
        "headers": [
            {"name": "X-SES-CONFIGURATION-SET", "value": "ConfigSet"},
            {"name": "From", "value": "sender@example.com"},
            {"name": "To", "value": "recipient@example.com"},
            {"name": "Subject", "value": "Amazon SES Test"}
        ]
    }
}
```
Failure Event Record

The following is an example of a failure event record that Amazon SES publishes to Amazon SNS.

```json
{
  "eventType": "Rendering Failure",
  "mail": {
    "timestamp": "2018-01-22T18:43:06.197Z",
    "source": "sender@example.com",
    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sendingAccount": "123456789012",
    "messageId": "EXAMPLE7c191be45-e9aed9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "destination": ["recipient@example.com"],
    "headersTruncated": false,
    "tags": {
      "ses:configuration-set": ["ConfigSet"]
    }
  },
  "failure": {
    "errorMessage": "Attribute 'attributeName' is not present in the rendering data.",
    "templateName": "MyTemplate"
  }
}
```
Amazon SES Event Publishing Tutorials

This section provides tutorials that demonstrate how to use Amazon SES event publishing with AWS services that enable you to analyze and visualize your data.

These tutorials are contained in the following sections.

- Analyze Email Sending Events With Amazon Redshift (p. 289)
- Visualize Email Sending Events With Amazon Elasticsearch Service and Kibana (p. 300)
- Graph Email Sending Events in CloudWatch (p. 310)
- Analyze Email Sending Events With Amazon Kinesis Data Analytics (p. 313)

Analyze Email Sending Events With Amazon Redshift

In this tutorial, you publish Amazon SES email sending events to an Amazon Kinesis Firehose delivery stream that publishes data to Amazon Redshift. You then connect to the Amazon Redshift database and use a SQL query tool to query the database for Amazon SES email sending events that meet certain criteria.

The following sections walk you through the process.

- Prerequisites (p. 289)
- Step 1: Create an Amazon Redshift Cluster (p. 290)
- Step 2: Connect to Your Amazon Redshift Cluster (p. 290)
- Step 3: Create a Database Table (p. 293)
- Step 4: Create a Kinesis Firehose Delivery Stream (p. 295)
- Step 5: Set up a Configuration Set (p. 298)
- Step 6: Send Emails (p. 298)
- Step 7: Query Email Sending Events (p. 299)

Prerequisites

For this tutorial, you will need the following:

- An AWS account – To access any web service that AWS offers, you must first create an AWS account at https://aws.amazon.com/.
- Verified email address – To send emails using Amazon SES, you must verify your "From" address or domain to show that you own it. If you are in the sandbox, you also must verify your "To" addresses. You can verify email addresses or entire domains, but this tutorial requires a verified email address so that you can send an email from the Amazon SES console, which is the simplest way to send an email. For information about how to verify an email address, see Verifying Email Addresses in Amazon SES (p. 43).
- A SQL query tool – Amazon Redshift does not provide or install any SQL client tools or libraries, so you must install one that you can use to access the Amazon Redshift clusters that contain your Amazon SES events. In this tutorial, we use SQL Workbench/J, a free, DBMS-independent, cross-platform SQL query tool. The following steps show you were to go to install it.

The following procedure shows how to install SQL Workbench/J.

**To install SQL Workbench/J on Your Client Computer**

1. Review the SQL Workbench/J software license.
2. Go to the SQL Workbench/J website and download the appropriate package for your operating system.

3. Go to Installing and starting SQL Workbench/J and install SQL Workbench/J.

   **Important**
   Note the Java runtime version prerequisites for SQL Workbench/J and ensure you are using that version. Otherwise, this client application will not run.

4. Go to Configure a JDBC Connection and download an Amazon Redshift JDBC driver to enable SQL Workbench/J to connect to your cluster.

Next Step

Step 1: Create an Amazon Redshift Cluster (p. 290)

**Step 1: Create an Amazon Redshift Cluster**

To create an Amazon Redshift cluster, go the Amazon Redshift console and choose Launch Cluster. A wizard guides you through choosing options for your cluster, and it provides default values for most options.

For this simple tutorial, type a cluster name and password, and then you can use all of the default values. You do not need to set any values specific to Amazon SES event publishing.

   **Important**
   The cluster that you deploy for this tutorial will run in a live environment. As long as it is running, it will accrue charges to your AWS account. To avoid unnecessary charges, you should delete your cluster when you are done with it. For pricing information, go to the Amazon Redshift pricing page.

Next Step

Step 2: Connect to Your Amazon Redshift Cluster (p. 290)

**Step 2: Connect to Your Amazon Redshift Cluster**

Now you will connect to your cluster by using a SQL client tool. For this tutorial, you use the SQL Workbench/J client that you installed in the prerequisites section (p. 289).

Complete this section by performing the following steps:

- **Getting Your Connection String (p. 290)**
- **Connecting to Your Cluster From SQL Workbench/J (p. 291)**

**Getting Your Connection String**

The following procedure shows how to get the connection string that you will need to connect to your Amazon Redshift cluster from SQL Workbench/J.

**To get your connection string**

1. In the Amazon Redshift console, in the navigation pane, choose Clusters.
2. To open your cluster, choose your cluster name.
3. On the Configuration tab, under Cluster Database Properties, copy the JDBC URL of the cluster.

   **Note**
   The endpoint for your cluster is not available until the cluster is created and in the available state.
Connecting to Your Cluster From SQL Workbench/J

The following procedure shows how to connect to your cluster from SQL Workbench/J. This procedure assumes that you installed SQL Workbench/J on your computer as described in Prerequisites (p. 289).

To connect to your cluster from SQL Workbench/J

1. Open SQL Workbench/J.
2. Choose File, and then choose Connect window.
3. Choose the Create a new connection profile button.
4. In the New profile text box, type a name for the profile.
5. At the bottom of the window, on the left, choose Manage Drivers.
6. In the Manage Drivers dialog box, choose the Create a new entry button, and then add the driver as follows.
a. In the **Name** box, type a name for the driver.
b. Next to **Library**, choose the folder icon.
c. Navigate to the location of the driver you downloaded in Configure a JDBC Connection, select the driver, and then choose **Open**.
d. Choose **OK**.

You will be taken back to the **Select Connection Profile** dialog box.

7. For **Driver**, choose the driver that you just added.
8. For **URL**, paste the JDBC URL that you copied from the Amazon Redshift console.
9. For **Username**, type the username that you chose when you set up the Amazon Redshift cluster (p. 290).
10. For **Password**, type the password that you chose when you set up the Amazon Redshift cluster.
11. Select **Autocommit**.
12. To test the connection, choose **Test**.

**Note**
If the connection attempt times out, you might need to add your IP address to the security group that allows incoming traffic from IP addresses. For more information, see The Connection Is Refused or Fails in the Amazon Redshift Database Developer Guide.
13. On the top menu bar, choose the **Save profile list** button.

14. Choose **OK**.

SQL Workbench/J will connect to your Amazon Redshift cluster.

**Next Step**

**Step 3: Create a Database Table (p. 293)**

**Step 3: Create a Database Table**

After you connect to the initial database in Amazon Redshift, you typically use the initial database as the base for creating a new database. However, in this simple tutorial, we create a table to hold your Amazon SES event publishing data directly within the initial database.

For this tutorial, let's assume that we're interested in the following fields within the email sending event records (p. 258). All of these fields, except for `mail.tags.campaign`, are provided automatically by Amazon SES. We introduce the `mail.tags.campaign` field when we send an email using `campaign` as a message tag in **Step 6: Send Emails (p. 298)**.

- `mail.messageId`
- `eventType`
- `mail.sendingAccountId`
- `mail.timestamp`
- `mail.destination`
- `mail.tags.ses:configuration-set`
- `mail.tags.campaign`
To access this information within your database, you must create a table. The following procedure shows how to specify this information when you create the table in your database.

**Note**
We assume that SQL Workbench/J is currently open on your computer, and it is connected to your Amazon Redshift cluster, as described in previous step (p. 290).

### To create a table using SQL Workbench/J

1. In SQL Workbench/J, copy the following code and paste it into the **Statement 1** window.

   ```sql
   create table ses (
       message_id varchar(200) not null,
       event_type varchar(20) not null,
       sending_account_id char(12),
       timestamp varchar(50),
       destination text,
       configuration_set text,
       campaign text
   );
   ```

2. Place the cursor within the statement (somewhere before the semicolon), and then choose the **Execute current statement** button, as shown in the following figure.

3. In the **Messages** pane, verify that your table was successfully created.

   **Next Step**

   **Step 4: Create a Kinesis Firehose Delivery Stream (p. 295)**
**Step 4: Create a Kinesis Firehose Delivery Stream**

To publish email sending events to Amazon Kinesis Firehose, you must create a Kinesis Firehose delivery stream. When you set up a Kinesis Firehose delivery stream, you choose where Kinesis Firehose publishes the data. For this tutorial, we will set up Kinesis Firehose to publish the data to Amazon Redshift, and choose to have Kinesis Firehose publish the records to Amazon S3 as an intermediary step. In the process, we need to specify how Amazon Redshift should copy records from Amazon S3 into the table we created in the previous step (p. 293).

This section shows how to create a Kinesis Firehose delivery stream that sends data to Amazon Redshift, and how to edit the delivery stream to specify how Amazon Redshift should copy the Amazon SES event publishing data to Amazon S3.

**Note**
You must have already set up the Amazon Redshift cluster (p. 290), connected to your cluster (p. 290), and created a database table (p. 293), as explained previous steps.

**Creating a Kinesis Firehose Delivery Stream**

The following procedure shows how to create a Kinesis Firehose delivery stream that publishes data to Amazon Redshift, using Amazon S3 as the intermediary data location.

**To create a delivery stream from Kinesis Firehose to Amazon Redshift**

1. Sign in to the AWS Management Console and open the Kinesis Firehose console at https://console.aws.amazon.com/firehose/.
2. Choose **Create Delivery Stream**.
3. On the **Destination** page, choose the following options.
   - **Destination** – Choose Amazon Redshift.
   - **Delivery stream name** – Type a name for the delivery stream.
   - **S3 bucket** – Choose **New S3 bucket**, type a bucket name, choose the region, and then choose **Create Bucket**.
   - **Redshift cluster** – Choose the Amazon Redshift cluster that you created in a previous step.
   - **Redshift database** – Type **dev**, which is the default database name.
   - **Redshift table** – Type **ses**, which is the table you created in Step 3: Create a Database Table (p. 293).
   - **Redshift table columns** – Leave this field empty.
   - **Redshift username** – Type the username that you chose when you set up the Amazon Redshift cluster (p. 290).
   - **Redshift password** – Type the password that you chose when you set up the Amazon Redshift cluster.
   - **Redshift COPY options** – Leave this field empty.
   - **Retry duration** – Leave this at its default value.
   - **COPY command** – Leave this at its default value. You will update it in the next procedure.
4. Choose **Next**.
5. On the **Configuration** page, leave the fields at the default settings for this simple tutorial. The only step you must do is select an IAM role that enables Kinesis Firehose to access your resources, as explained in the following procedure.
   a. For **IAM Role**, choose **Select an IAM role**.
   b. In the drop-down menu, under **Create/Update existing IAM role**, choose **Firehose delivery IAM role**.
You will be taken to the IAM console.

c. In the IAM console, leave the fields at their default settings, and then choose Allow.

![IAM Console Screenshot]

You will return to the Kinesis Firehose delivery stream set-up steps in the Kinesis Firehose console.

6. Choose Next.

7. On the Review page, review your settings, and then choose Create Delivery Stream.

**Setting Amazon Redshift Copy Options**

Next, you must specify to Amazon Redshift how to copy the Amazon SES event publishing JSON records into the database table you created in Step 3: Create a Database Table (p. 293). You do this by editing the copy options in the Kinesis Firehose delivery stream.

For this procedure, you must create a JSONPaths file. A JSONPaths file is a text file that specifies to the Amazon Redshift COPY command how to parse the JSON source data. We provide a JSONPaths file in the procedure. For more information about JSONPaths files, see COPY from JSON Format in the Amazon Redshift Database Developer Guide.

You upload the JSONPaths file to the Amazon S3 bucket you set up when you created the Kinesis Firehose delivery stream, and then edit the COPY options of the Kinesis Firehose delivery stream to use the JSONPaths file you uploaded. These steps are explained in the following procedure.

**To set Amazon Redshift COPY command options**

1. Create a JSONPaths file – On your computer, create a file called jsonpaths.json. Copy the following text into the file, and then save the file.

```json
{
    "jsonpaths": [
```
2. **Upload the JSONPaths file to the Amazon S3 bucket** – Go to the Amazon S3 console and upload the file to the bucket you created when you set up the Kinesis Firehose delivery stream in *Creating a Kinesis Firehose Delivery Stream* (p. 295).

3. **Set the COPY command in the Kinesis Firehose delivery stream settings** – Now you have the information you need to set the syntax of the COPY command that Amazon Redshift uses when it puts your data in the table you created. The following procedure shows how to update the COPY command information in the Kinesis Firehose delivery stream settings.

   1. Go to the Kinesis Firehose console.
   2. Under Redshift Delivery Streams, choose the Kinesis Firehose delivery stream that you created for Amazon SES event publishing.
   3. On the Details page, choose Edit.
   4. In the Redshift COPY options box, type the following text, replacing the following values with your own values:
      - **S3-BUCKET-NAME** – The name of the Amazon S3 bucket where Kinesis Firehose places your data for Amazon Redshift to access. You created this bucket when you set up your Kinesis Firehose delivery stream in *Step 4: Create a Kinesis Firehose Delivery Stream* (p. 295). An example is my-bucket.
      - **REGION** – The region in which your Amazon SES, Kinesis Firehose, Amazon S3, and Amazon Redshift resources are located. An example is us-west-2.

      ```
      json 's3://S3-BUCKET-NAME/jsonpaths.json' region 'REGION';
      ```
   5. Choose Save.
Next Step

Step 5: Set up a Configuration Set (p. 298)

Step 5: Set up a Configuration Set

To set up Amazon SES to publish your email sending events to Amazon Kinesis Firehose, you first create a configuration set, and then you add a Kinesis Firehose event destination to the configuration set. This section shows how to accomplish those tasks.

If you already have a configuration set, you can add a Kinesis Firehose destination to your existing configuration set. In this case, skip to Adding a Kinesis Firehose Event Destination (p. 298).

Creating a Configuration Set

The following procedure shows how to create a configuration set.

To create a configuration set

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose Configuration Sets.
3. In the content pane, choose Create Configuration Set.
4. Type a name for the configuration set, and then choose Create Configuration Set.
5. Choose Close.

Adding a Kinesis Firehose Event Destination

The following procedure shows how to add a Kinesis Firehose event destination to the configuration set you created.

To add a Kinesis Firehose event destination to the configuration set

1. Choose the configuration set from the configuration set list.
2. For Add Destination, choose Select a destination type, and then choose Kinesis Firehose.
3. For Name, type a name for the event destination.
4. Select all Event types.
5. Select Enabled.
6. For Stream, choose the delivery stream that you created in Step 4: Create a Kinesis Firehose Delivery Stream (p. 295).
7. For IAM role, choose Let SES make a new role, and then type a name for the role.
8. Choose Save.
9. To exit the Edit Configuration Set page, use the back button of your browser.

Next Step

Step 6: Send Emails (p. 298)

Step 6: Send Emails

For Amazon SES to publish events associated with an email, you must specify a configuration set when you send the email. You can also include message tags to categorize the email. This section shows how to send a simple email that specifies a configuration set and message tags using the Amazon
To send an email using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation pane of the Amazon SES console, under Identity Management, choose Email Addresses.
3. In the list of identities, select the check box of an email address that you have successfully verified with Amazon SES (p. 43).
4. Choose Send a Test Email.
5. In the Send Test Email dialog box, for Email Format, choose Raw.
6. For the To address, type an address from the Amazon SES mailbox simulator (p. 162), such as complaint@simulator.amazonses.com or bounce@simulator.amazonses.com.
7. Copy and paste the following message in its entirety into the Message text box, replacing CONFIGURATION-SET-NAME with the name of the configuration set you created in Step 5: Set up a Configuration Set (p. 298), and replacing FROM-ADDRESS with the verified address you are sending this email from.

   X-SES-MESSAGE-TAGS: campaign=book
   X-SES-CONFIGURATION-SET: CONFIGURATION-SET-NAME
   Subject: Amazon SES Event Publishing Test
   From: Amazon SES User <FROM-ADDRESS>
   MIME-Version: 1.0
   Content-Type: text/plain
   This is a test message.

8. Choose Send Test Email.
9. Repeat this procedure a few times so that you generate multiple email sending events. For a few of the emails, change the value of the campaign message tag to clothing to simulate sending for a different email campaign. That way, when you query your Amazon Redshift database for email sending event records in the last step of this tutorial, you can experiment with querying based on email campaign.

Next Step
Step 7: Query Email Sending Events (p. 299)

Step 7: Query Email Sending Events

Now that you have generated some email sending events by sending emails with your configuration set and message tags, you can query those records in Amazon Redshift.

Note
We assume that SQL Workbench/J is currently open on your computer, and it is connected to your Amazon Redshift cluster, as described in Step 2: Connect to Your Amazon Redshift Cluster (p. 290).

To query email sending event data in Amazon Redshift from SQL Workbench/J

1. To display all of your email sending records, copy the following query and paste it into the Statement 1 window.

   select * from ses;
2. Place the cursor within the statement (somewhere before the semicolon), and then choose the **Execute current statement** button.

You will see the email sending records for all of the emails you sent in Step 6: Send Emails (p. 298). The records in the following figure show that our book campaign had two complaints, and the clothing campaign had one bounce.

3. To count the complaint records for the campaign of type book, copy the following query and paste it into the **Statement 1** window.

```sql
select count(*) as numberOfComplaint from ses where event_type = 'Complaint' and campaign like '%book%';
```

4. Place the cursor within the statement (somewhere before the semicolon), and then choose the **Execute current statement** button.

The results are the following, showing that the book campaign had two complaints.

### Visualize Email Sending Events With Amazon Elasticsearch Service and Kibana

Elasticsearch is an open-source search and analytics engine for use cases such as log analytics and real-time application monitoring. Amazon Elasticsearch Service (Amazon ES) is an AWS service that enables you to deploy, operate, and scale Elasticsearch in the AWS cloud. You can use Amazon ES to analyze your Amazon SES email sending events.

In this tutorial, you publish Amazon SES email sending events to an Amazon Kinesis Firehose delivery stream that publishes the event data to Amazon ES. You then view the data with **Kibana**, an open-source visualization tool designed to work with Elasticsearch. Amazon ES includes built-in integration with Kibana.
The following sections walk you through the process.

- **Prerequisites (p. 301)**
- **Step 1: Create an Amazon ES Cluster (p. 301)**
- **Step 2: Create a Kinesis Firehose Delivery Stream (p. 302)**
- **Step 3: Set up a Configuration Set (p. 304)**
- **Step 4: Send Emails (p. 304)**
- **Step 5: Visualize Data in Kibana (p. 305)**

**Prerequisites**

For this tutorial, you will need the following:

- **An AWS account** – To access any web service that AWS offers, you must first create an AWS account at [https://aws.amazon.com/](https://aws.amazon.com/).

- **Verified email address** – To send emails using Amazon SES, you must verify your "From" address or domain to show that you own it. If you are in the sandbox, you also must verify your "To" addresses. You can verify email addresses or entire domains, but this tutorial requires a verified email address so that you can send an email from the Amazon SES console, which is the simplest way to send an email. For information about how to verify an email address, see Verifying Email Addresses in Amazon SES (p. 43).

**Next Step**

**Step 1: Create an Amazon ES Cluster (p. 301)**

**Step 1: Create an Amazon ES Cluster**

Before you set up Amazon Kinesis Firehose to publish your Amazon SES email sending events to Amazon Elasticsearch Service (Amazon ES), you must create an Amazon ES cluster. This section shows how to create an Amazon ES cluster using the Amazon ES console.

For the simplicity of this tutorial, we choose basic options. For information about all available options, see the Amazon Elasticsearch Service Developer Guide.

**Important**

The cluster that you deploy for this tutorial will run in a live environment. As long as it is running, it will accrue charges to your AWS account. To avoid unnecessary charges, you should delete your cluster when you are done with it. For pricing information, go to the Amazon Elasticsearch Service pricing page.

**To create an Amazon ES cluster**

2. In the Amazon ES console, choose Get started.
3. On the Define domain page, under Domain Name, type a name for your Amazon ES domain.
4. Under Version, leave the Elasticsearch version field at its default value.
5. Choose Next.
6. On the Configure cluster page, under Node configuration, choose the following options.
   - **Instance count** – Type 1.
   - **Instance type** – Choose t2.micro.elasticsearch (Free tier eligible).
   - **Enable dedicated master** – Do not enable this option.
7. Under Storage configuration, choose the following options.

- **Enable zone awareness** – Do not enable this option.
- **Storage type** – Choose EBS. For the EBS settings, choose **EBS volume type** of General Purpose (SSD) and **EBS volume size** of 10.
- **Automated snapshot start hour** – Choose **Automated snapshots start hour 00:00 UTC (default)**.

8. Choose **Next**.

9. On the **Set up access policy** page, for **Set the domain access policy to**, choose **Allow open access to the domain**.

   **Important**
   This setting simplifies testing but is **not recommended for production environments**. For information about configuring access policies, see Configuring Access Policies in the Amazon Elasticsearch Service Developer Guide.

10. Choose **Next**.

11. On the **Review** page, review your settings, and then choose **Confirm and create**.

   **Note**
   The cluster will take up to ten minutes to deploy.

Next Step

**Step 2: Create a Kinesis Firehose Delivery Stream (p. 302)**

**Step 2: Create a Kinesis Firehose Delivery Stream**

To publish email sending events to Amazon Kinesis Firehose, you must create a Kinesis Firehose delivery stream. When you set up a Kinesis Firehose delivery stream, you choose where Kinesis Firehose publishes the data. In this tutorial, we set up Kinesis Firehose to publish the data to Amazon Elasticsearch Service (Amazon ES).

This section shows how to create a Kinesis Firehose delivery stream using the Kinesis Firehose console. For the simplicity of this tutorial, we choose basic options. For information about all available options, see Creating an Amazon Kinesis Firehose Delivery Stream in the Amazon Kinesis Data Firehose Developer Guide.

**Note**
You must have already set up an Amazon ES cluster, as explained in **Step 1: Create an Amazon ES Cluster (p. 301)**.

**To create a delivery stream from Kinesis Firehose to Amazon Elasticsearch Service**

1. Sign in to the AWS Management Console and open the Kinesis Firehose console at https://console.aws.amazon.com/firehose/.
2. Choose **Create Delivery Stream**.
3. On the **Destination** page, choose the following options.

   - **Destination** – Choose Amazon Elasticsearch Service.
   - **Delivery stream name** – Type a name for the delivery stream.
   - **Elasticsearch domain** – Choose the Amazon ES domain that you created in Step 1: Create an Amazon ES Cluster (p. 301).
   - **Index** – Type a name that you want to use to explore your email sending event data in Kibana. You can choose any name, but let's use **holiday-sale** for this tutorial. An index is analogous to a database. For example, if you want an easy way to access events from each of your email campaigns separately, you can use a different Kinesis Firehose stream and index for each campaign.
• **Index rotation** – Choose NoRotation.

• **Type** – Although this setting is not relevant to this tutorial, you must choose something, so type events. A type is a logical category or partition of your index.

• **Retry duration (sec)** – Type 300.

• **Backup mode** – Choose Failed Documents Only.

• **S3 bucket** – Choose New S3 Bucket. Type a name for the bucket and choose the region your console is currently using.

• **S3 prefix** – Leave this field empty.

4. Choose Next.

5. On the **Configuration** page, leave the fields at the default settings. The only step you must do is select an IAM role that enables Kinesis Firehose to access your resources, as explained in the following procedure.

   a. For **IAM Role**, choose **Select an IAM role**.

   b. In the drop-down menu, under **Create/Update existing IAM role**, choose **Firehose delivery IAM role**.

      You will be taken to the IAM console.

   c. In the IAM console, leave the fields at their default settings, and then choose **Allow**.

6. Choose Next.

7. On the **Review** page, review your settings, and then choose **Create Delivery Stream**.

**Next Step**

**Step 3: Set up a Configuration Set (p. 304)**
Creating a Configuration Set

The following procedure shows how to create a configuration set.

To create a configuration set

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose **Configuration Sets**.
3. In the content pane, choose **Create Configuration Set**.
4. Type a name for the configuration set, and then choose **Create Configuration Set**.
5. Choose **Close**.

Adding a Kinesis Firehose Event Destination

The following procedure shows how to add a Kinesis Firehose event destination to the configuration set you created.

To add a Kinesis Firehose event destination to the configuration set

1. Choose the configuration set from the configuration set list.
2. For **Add Destination**, choose **Select a destination type**, and then choose **Kinesis Firehose**.
3. For **Name**, type a name for the event destination.
4. Select all **Event types**.
5. Select **Enabled**.
6. For **Stream**, choose the delivery stream that you created in Step 2: Create a Kinesis Firehose Delivery Stream (p. 302).
7. For **IAM role**, choose **Let SES make a new role**, and then type a name for the role.
8. Choose **Save**.
9. To exit the **Edit Configuration Set** page, use the back button of your browser.
To send an email using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the Navigation pane of the Amazon SES console, under Identity Management, choose Email Addresses.
3. In the list of identities, select the check box of an email address that you have successfully verified with Amazon SES (p. 43).
4. Choose Send a Test Email.
5. In the Send Test Email dialog box, for Email Format, choose Raw.
6. For the To address, type an address from the Amazon SES mailbox simulator (p. 162), such as complaint@simulator.amazonses.com or bounce@simulator.amazonses.com.
7. Copy and paste the following message in its entirety into the Message text box, replacing CONFIGURATION-SET-NAME with the name of the configuration set you created in Step 3: Set up a Configuration Set (p. 304), and replacing FROM-ADDRESS with the verified address you are sending this email from.

```
X-SES-MESSAGE-TAGS: campaign=book
X-SES-CONFIGURATION-SET: CONFIGURATION-SET-NAME
Subject: Amazon SES Event Publishing Test
From: Amazon SES User <FROM-ADDRESS>
MIME-Version: 1.0
Content-Type: text/plain
This is a test message.
```
8. Choose Send Test Email.
9. Repeat this procedure a few times so that you generate multiple email sending events. For a few of the emails, change the value of the campaign message tag to clothing to simulate sending for a different email campaign.

Next Step

Step 5: Visualize Data in Kibana (p. 305)

Step 5: Visualize Data in Kibana

Now that you have published some Amazon SES email sending events to Amazon Elasticsearch Service (Amazon ES) by sending emails with your configuration set and message tags, you can visualize the events using Kibana, a web interface for Elasticsearch.

This section shows how to find your email sending events in Kibana, graph your email sending events by event type, and find the email addresses that bounced. These exercises are useful because monitoring your bounces and complaints is an important part of maintaining your mailing list.

- Viewing Your Email Sending Events (p. 305)
- Graphing Your Email Sending Events by Type (p. 307)
- Finding Recipient Addresses That Bounced (p. 308)

For Kibana documentation and tutorials, see the Kibana User Guide.

Viewing Your Email Sending Events

The following procedure shows how to go to the Kibana user interface from the Amazon ES console. It then shows how to view the email sending events associated with the index you defined when you
set up your Amazon Kinesis Firehose delivery stream in Step 2: Create a Kinesis Firehose Delivery Stream (p. 302).

To view your raw email sending events in Kibana

1. Sign in to the AWS Management Console and open the Amazon Elasticsearch Service console at https://console.aws.amazon.com/es/.
2. Under My Elasticsearch domains, choose the domain you created in Step 1: Create an Amazon ES Cluster (p. 301).
3. Choose the Kibana link.

You will be directed to the Kibana web interface.

4. On the Configure an index pattern page, clear the Index contains time-based events check box.
5. Under Index name or pattern, verify that holiday-sale, the index you created in Step 1: Create an Amazon ES Cluster (p. 301), is present. If it is not present, type holiday-sale* into the field, and then choose Create.

   **Note**
   If the Create button does not appear, try adding an asterisk to the end of the index pattern.

6. Choose the Discover tab on the top menu.
7. In the search box below the Discover tab, put the cursor after the asterisk (*), and then press Enter.

   Kibana will display a list of all of your email sending events.
Graphing Your Email Sending Events by Type

For insight into the overall health of your email campaign, you can visually compare the number of problematic outcomes (bounces, complaints, and rejected emails) you have received across your campaign. The following procedure shows how to set up a vertical bar chart that displays the count of each type of email sending event.

To graph your email sending events by type

1. Choose the Visualize tab on the top menu.
2. Choose Vertical bar chart.
3. Choose From a new search. If prompted for an index pattern, choose holiday-sale*.
4. On the metrics pane, next to Y-Axis, ensure that the metric is set to Count.
5. In the Buckets pane, choose X-Axis.
6. For Aggregation, choose Terms. Terms refers to the fields in your JSON documents in your index.
7. For Field, under String, choose eventType. Leave the rest of the fields at their default values.
8. Next to Options, choose the play button. Your bar chart comparing the event types will display on the screen.
9. To save your visualization, choose the save icon from the group of icons to the right of the search bar.

10. Type a title such as All Event Types, and then choose Save.

Finding Recipient Addresses That Bounced

Now that you have a visualization of the health of your overall email campaign, you can go back to the raw data and find out which recipient addresses bounced, as described in the following procedure.

To view the email addresses that bounced

1. Go to the Discover tab on the top menu.

   You will see a list of your raw event records.

   **Note**
   If the main window reports that there are zero search results, enter * in the search bar, and then press Enter.

2. In the left pane, under Available Fields, hover over eventType, and then choose the add button that appears next to it.
3. In the main window, hover over the **eventType** column heading, and then choose the arrow to sort the event types by name.

The bounce events will move to the top of the list.

**Note**
There might be a short delay before the events are resorted.

4. In the left pane, under **Available Fields**, hover over **bounce.bouncedRecipients**, and then choose the **add** button that appears next to it.

In the main window, you will see the recipient address and bounce reason for each bounce event.
Graph Events in Amazon CloudWatch

In this tutorial, you publish Amazon SES email sending events to Amazon CloudWatch and then graph the events using the CloudWatch console.

The following sections walk you through the process.

- Prerequisites (p. 310)
- Step 1: Set up a Configuration Set (p. 310)
- Step 2: Send Emails (p. 312)
- Step 3: Graph Events (p. 313)

Prerequisites

For this tutorial, you will need the following:

- **An AWS account** – To access any web service that AWS offers, you must first create an AWS account at https://aws.amazon.com/.
- **Verified email address** – To send emails using Amazon SES, you must verify your "From" address or domain to show that you own it. If you are in the sandbox, you also must verify your "To" addresses. You can verify email addresses or entire domains, but this tutorial requires a verified email address so that you can send an email from the Amazon SES console, which is the simplest way to send an email. For information about how to verify an email address, see Verifying Email Addresses in Amazon SES (p. 43).

Next Step

**Step 1: Set up a Configuration Set (p. 310)**

**Step 1: Set up a Configuration Set**

To set up Amazon SES to publish your email sending events to Amazon CloudWatch, you first create a configuration set, and then you add a CloudWatch event destination to the configuration set. This section shows how to accomplish those tasks.
If you already have a configuration set, you can add a CloudWatch destination to your existing configuration set. In this case, skip to Adding a CloudWatch Event Destination (p. 311).

Creating a Configuration Set

The following procedure shows how to create a configuration set.

To create a configuration set
1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose Configuration Sets.
3. In the content pane, choose Create Configuration Set.
4. Type a name for the configuration set, and then choose Create Configuration Set.
5. Choose Close.

Adding a CloudWatch Event Destination

The following procedure shows how to add a CloudWatch event destination to the configuration set you created.

To add a CloudWatch event destination to a configuration set
1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose Configuration Sets.
3. Choose the configuration set you created in the previous procedure.
4. For Add Destination, choose Select a destination type, and then choose CloudWatch.
5. For Name, type a name for the event destination.
6. For Event types, choose Bounce and Complaint.
7. Select Enabled.
8. For Value Source, choose Message Tag.
9. For Dimension Name, type campaign.
10. For Default Value, type unknown.
11. Choose Save.
12. To exit the Edit Configuration Set page, use the back button of your browser.

**Step 2: Send Emails**

For Amazon SES to publish events associated with an email, you must specify a configuration set when you send the email. You can also include message tags to categorize the email. This section shows how to send a simple email that specifies a configuration set and message tags using the Amazon SES console. You send the email to the Amazon SES mailbox simulator so that you can test bounces, complaints, and other email sending outcomes.

**To send an email using the Amazon SES console**

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the Navigation pane of the Amazon SES console, under Identity Management, choose Email Addresses.
3. In the list of identities, select the check box of an email address that you have successfully verified with Amazon SES (p. 43).
4. Choose Send a Test Email.
5. In the Send Test Email dialog box, for Email Format, choose Raw.
6. For the To address, type an address from the Amazon SES mailbox simulator (p. 162), such as complaint@simulator.amazonses.com or bounce@simulator.amazonses.com.
7. Copy and paste the following message in its entirety into the Message text box, replacing CONFIGURATION-SET-NAME with the name of the configuration set you created in Step 3: Set up a Configuration Set (p. 304), and replacing FROM-ADDRESS with the verified address you are sending this email from.

```plaintext
X-SES-MESSAGE-TAGS: campaign=book
X-SES-CONFIGURATION-SET: CONFIGURATION-SET-NAME
Subject: Amazon SES Event Publishing Test
From: Amazon SES User <FROM-ADDRESS>
MIME-Version: 1.0
```
This is a test message.

8. Choose Send Test Email.
9. Repeat this procedure a few times so that you generate multiple email sending events. For a few of the emails, change the value of the campaign message tag to clothing to simulate sending for a different email campaign.

Next Step

Step 3: Graph Email Sending Events (p. 313)

Step 3: Graph Email Sending Events

Now that you have published some Amazon SES email sending events to CloudWatch by sending emails with your configuration set and message tags, you can graph metrics for those events using the CloudWatch console.

**To graph email sending event metrics**

1. Sign in to the AWS Management Console and open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the left navigation pane, choose Metrics.
3. In the All metrics tab, choose SES.
4. Choose Per configuration set.
5. Select the metric to view.

A graph appears in the details pane.

Analyze Email Sending Events With Amazon Kinesis Data Analytics

Amazon Kinesis Data Analytics enables you to process and analyze streaming data using SQL. You can use Amazon Kinesis Data Analytics to analyze your Amazon SES email sending events.

In this tutorial, you first set up an Amazon SES configuration set to publish your email sending events to an Amazon Kinesis Firehose delivery stream, and then you send emails through Amazon SES using that configuration set. You then set up Amazon Kinesis Data Analytics to capture the email sending events from the Kinesis Firehose stream and use SQL to extract key information from the emails you sent.

**Note**

This tutorial requires that you have an application that can send a steady stream of emails through Amazon SES. This requirement is explained in Prerequisites (p. 314).

The following sections walk you through the tutorial.

- Prerequisites (p. 314)
- Step 1: Create a Kinesis Firehose Delivery Stream (p. 314)
- Step 2: Set up a Configuration Set (p. 315)
- Step 3: Send Emails (p. 316)
- Step 4: Create an Amazon Kinesis Data Analytics Application (p. 317)
- Step 5: Run a SQL Query (p. 321)
• (Optional) Step 6: Save SQL Query Results (p. 322)

Prerequisites

For this tutorial, you need the following:

• **An AWS account** – To access any web service that AWS offers, you must first create an AWS account at https://aws.amazon.com/.

• **Verified email address** – To send emails using Amazon SES, you must verify your "From" address or domain to show that you own it. If you are in the sandbox, you also must verify your "To" addresses. You can verify email addresses or entire domains, but this tutorial requires a verified email address so that you can send an email from the Amazon SES console, which is the simplest way to send an email. For information about how to verify an email address, see Verifying Email Addresses in Amazon SES (p. 43).

• **Email application** – To use Amazon Kinesis Data Analytics as described in this tutorial, you must send a steady stream of emails through Amazon SES so that you generate a steady stream of email sending events. This enables Amazon Kinesis Data Analytics to automatically detect the schema and then to process the event records with SQL. Sending one email every ten seconds for five minutes is sufficient for this tutorial.

  **Important**
  If you do not have an existing email campaign to send to real recipients, we strongly recommend that you send emails to an Amazon SES mailbox simulator (p. 162) address. Emails that you send to the mailbox simulator do not count toward your Amazon SES bounce and complaint rates or your daily sending quota.

Next Step

Step 1: Create a Kinesis Firehose Delivery Stream (p. 314)

Step 1: Create a Kinesis Firehose Delivery Stream

To analyze Amazon SES email sending events with Amazon Kinesis Data Analytics, you must configure Amazon SES to publish the events to an Amazon Kinesis Firehose delivery stream, and then configure Amazon Kinesis Data Analytics to get the event data from Kinesis Firehose.

When you set up a Kinesis Firehose delivery stream, you choose the final destination of the data. Your destination options are Amazon Simple Storage Service (Amazon S3), Amazon Elasticsearch Service, and Amazon Redshift. If you simply want to analyze email sending events with Amazon Kinesis Data Analytics, it does not matter which destination you choose. For this tutorial, we configure Kinesis Firehose to publish the data to Amazon S3, but you can use the other destination options if they are in the same region as your Amazon SES sending and Kinesis Firehose delivery stream.

This section shows how to create a Kinesis Firehose delivery stream using the Kinesis Firehose console. For this tutorial, we choose basic options. For information about all available options, see Creating an Amazon Kinesis Firehose Delivery Stream in the Amazon Kinesis Data Firehose Developer Guide.

To create a delivery stream from Kinesis Firehose to Amazon S3

1. Sign in to the AWS Management Console and open the Kinesis Firehose console at https://console.aws.amazon.com/firehose/.
2. Choose Create Delivery Stream.
3. On the Destination page, choose the following options.
   • **Destination** – Choose Amazon S3.
   • **Delivery stream name** – Type a name for the delivery stream.
• **S3 bucket** – Choose an existing bucket, or choose New S3 Bucket. If you create a new bucket, type a name for the bucket and choose the region your console is currently using.

• **S3 prefix** – Leave this field empty.

4. Choose Next.

5. On the **Configuration** page, leave the fields at the default settings. The only required step is to select an IAM role that enables Kinesis Firehose to access your resources, as follows:

   a. For **IAM Role**, choose Select an IAM role.

   b. In the drop-down menu, under Create/Update existing IAM role, choose Firehose delivery IAM role.

      You are taken to the IAM console.

   c. In the IAM console, leave the fields at their default settings, and then choose Allow.

   ![Image of IAM console with Allow button highlighted]

   You return to the Kinesis Firehose delivery stream set-up steps in the Kinesis Firehose console.

6. Choose Next.

7. On the **Review** page, review your settings, and then choose Create Delivery Stream.

**Next Step**

**Step 2: Set up a Configuration Set (p. 315)**

**Step 2: Set up a Configuration Set**

To set up Amazon SES to publish your email sending events to Amazon Kinesis Firehose, you create a configuration set, and then you add a Kinesis Firehose event destination to the configuration set. This section describes how to accomplish those tasks.

If you already have a configuration set, you can add a Kinesis Firehose destination to your existing configuration set. In this case, skip to Adding a Kinesis Firehose Event Destination (p. 316).
Creating a Configuration Set

The following procedure describes how to create a configuration set.

To create a configuration set

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the left navigation pane, choose Configuration Sets.
3. In the content pane, choose Create Configuration Set.
4. Type a name for the configuration set, and then choose Create Configuration Set.
5. Choose Close.

Adding a Kinesis Firehose Event Destination

The following procedure shows how to add a Kinesis Firehose event destination to the configuration set you created.

To add a Kinesis Firehose event destination to the configuration set

1. Choose the configuration set from the configuration set list.
2. For Add Destination, choose Select a destination type, and then choose Kinesis Firehose.
3. For Name, type a name for the event destination.
4. Select all Event types.
5. Select Enabled.
6. For Stream, choose the delivery stream that you created in Step 1: Create a Kinesis Firehose Delivery Stream (p. 314).
7. For IAM role, choose Let SES make a new role, and then type a name for the role.
8. Choose Save.
9. To exit the Edit Configuration Set page, use the back button of your browser.

Next Step

Step 3: Send Emails (p. 316)

Step 3: Send Emails

Because this tutorial uses the Amazon Kinesis Data Analytics console to process and analyze streaming data, you must set up a steady stream of emails through Amazon SES. This tutorial assumes that you have an application that can send these emails. Sending one email every ten seconds for five minutes is sufficient for this tutorial. We highly recommend that you use a "To" address from the Amazon SES mailbox simulator (p. 162), such as success@simulator.amazonses.com.

To enable event publishing for an email, you provide the name of the configuration set to Amazon SES when you send the email. You can optionally include message tags to categorize the email. You provide this information to Amazon SES as either parameters to the email sending API, Amazon SES-specific email headers, or custom headers in your MIME message. For more information, see Send Email Using Amazon SES Event Publishing (p. 255).

For example, you might add the following Amazon SES-specific email headers to your email to simulate a book campaign. Replace CONFIGURATION-SET-NAME with the name of the configuration set you created in Step 2: Set up a Configuration Set (p. 315).
Next Step

Step 4: Create an Amazon Kinesis Data Analytics Application (p. 317)

Step 4: Create an Amazon Kinesis Data Analytics Application

Now that you have set up event publishing with Amazon SES, you can configure Amazon Kinesis Data Analytics to capture the email sending event data from your Amazon Kinesis Firehose delivery stream. To do this, you create an Amazon Kinesis Data Analytics application.

The following procedure shows how to use the Amazon Kinesis Data Analytics console to create an application that captures Amazon SES email sending event data from your Kinesis Firehose delivery stream, and then how to perform a simply SQL query on the data to return the events of type "Send".

Note
The email sending events of different event types (send, bounce, complaint, and delivery) have different JSON schemas (p. 258). In a production environment, you might examine several fields of this schema, but in this tutorial, we limit our examination to a small set of fields that are present for all event types.

To create an Amazon Kinesis Data Analytics application

1. Start sending a steady stream of emails configured for event publishing through Amazon SES, and continue sending the emails throughout this procedure. This is required so that Amazon Kinesis Data Analytics can automatically detect the schema of the event records. Sending one email every ten seconds for five minutes is sufficient for this tutorial. For more information, see Step 3: Send Emails (p. 316).

After your email program has sent a few emails, move to the next step.
2. Sign in to the AWS Management Console and open the Kinesis Data Analytics console at https://console.aws.amazon.com/kinesisanalytics.
3. Choose Create new application.
4. Enter an application name and description, and then choose Save and continue.
5. Choose Connect to a source.
6. Choose the Kinesis Firehose stream you created in Step 2: Set up a Configuration Set (p. 315).

Amazon Kinesis Data Analytics attempts to discover the schema of the email sending event records based on the incoming records. If Amazon Kinesis Data Analytics displays Error discovering input schema, that means that Amazon Kinesis Data Analytics has not received any email sending records yet. Choose Rediscover schema. You might need to choose this button several times. If schema discovery does not succeed after several attempts, ensure that your email sending application is steadily sending emails, and that the emails specify a configuration set.

When Amazon Kinesis Data Analytics detects a schema, it displays a success message and lists the records it detected.

Important
Do not choose Save and continue. This will cause errors because the discovered schema does not adhere to SQL naming constraints. You must edit the schema as described in the next step.

7. Choose Edit schema.
8. For this tutorial, we remove most of the rows. Choose X next to all rows except rows with the following column names:

- eventType
- timestamp
- messageId
- to
- ses:configuration-set

**Important**
Do not choose *Save schema and update stream samples*. This will cause errors because the discovered schema does not adhere to SQL naming constraints. You must edit the schema as described in the next step.
9. Examine the remaining entries under **Column name** and compare them to the SQL naming requirements as follows:

- **Format** – As described in **Identifiers** in the *Amazon Kinesis Data Analytics SQL Reference*, unquoted identifiers must start with a letter or underscore, and be followed by letters, digits, or underscores. Amazon SES auto-tag names do not comply with these requirements because they contain colons and dashes. You will edit these in the next step.

- **Reserved words** – Column names must not conflict with the SQL reserved words listed in **Reserved Words and Keywords** in the *Amazon Kinesis Data Analytics SQL Reference*. Examples of reserved keywords that conflict with Amazon SES event records are **timestamp**, **value**, **date**, **from**, and **to**.

10. Edit the remaining column names to conform to the SQL requirements as follows:

- Rename **ses:configuration-set** to **ses_configuration_set**.
- Rename **timestamp** to **ses_timestamp**.
- Rename **to** to **ses_to**.
11. Choose **Save schema and update stream samples**. If you encounter validation errors, ensure that you correctly performed step 10. If you encounter the **No rows in source stream** error, ensure that you are still sending the email stream that you started at the beginning of this procedure, and then choose **Retrieve rows**. You might need to choose **Retrieve rows** several times before Amazon Kinesis Data Analytics captures records.

12. Upon successful retrieval of rows, choose **Exit (done)**.
Step 5: Run a SQL Query

Now that you have created an Amazon Kinesis Data Analytics application and configured it to use your Amazon Kinesis Firehose delivery stream as its source, you can query the email sending event data that the Kinesis Firehose delivery stream receives.

This topic shows how to run a SQL query on the email sending event data.

**Important**
This procedure requires that you continue to send a steady stream of emails configured for event publishing through Amazon SES, as described in Step 3: Send Emails (p. 316).

To run a SQL query in Amazon Kinesis Data Analytics

1. Assuming that you have moved on to this procedure after completing the last step (p. 317), go to the Amazon Kinesis Data Analytics console top menu and choose your application.

2. Choose **Go to SQL editor**.

   Amazon Kinesis Data Analytics attempts to read event data from the Kinesis Firehose stream. If you encounter the **No rows in source stream** error, ensure that you are still sending the email stream you started at the beginning of this procedure, and then choose **Retrieve rows**.

3. In the code editor box, paste the following.
CREATE OR REPLACE STREAM "DESTINATION_SQL_STREAM" ("eventType" VARCHAR(16), "ses_timestamp" timestamp, "messageId" VARCHAR(64), "ses_to" VARCHAR(64), "ses_configuration_set" VARCHAR(32));

CREATE OR REPLACE PUMP "STREAM_PUMP" AS INSERT INTO "DESTINATION_SQL_STREAM"
SELECT STREAM "eventType", "ses_timestamp", "messageId", "ses_to", "ses_configuration_set"
FROM "SOURCE_SQL_STREAM_001"
WHERE "eventType" = 'Send'

4. Choose **Save and run SQL**.

After Amazon Kinesis Data Analytics retrieves and processes incoming records, you see a list of event records of type "Send".

Next Step

(Optional) Step 6: Save SQL Query Results (p. 322)

(Optional) Step 6: Save SQL Query Results

You can set up your Amazon Kinesis Data Analytics application to write the output of your SQL queries to an Amazon Kinesis Firehose delivery stream. To do so, you must create another Kinesis Firehose delivery stream because you cannot use the same delivery stream as both the source and destination of an Amazon Kinesis Data Analytics application. As with any Kinesis Firehose delivery stream, you can choose Amazon Simple Storage Service (Amazon S3), Amazon Elasticsearch Service, or Amazon Redshift as the destination.

The following procedure shows how to configure Amazon Kinesis Data Analytics to save SQL query results in JSON format to a Kinesis Firehose delivery stream that writes the data to Amazon S3. Then you run a SQL query and access the saved data.
To save the results of SQL queries to Amazon S3

1. Set up a new Kinesis Firehose stream that uses Amazon S3 as the destination. It is the same procedure as Step 1: Create a Kinesis Firehose Delivery Stream (p. 314).

2. Go to the Amazon Kinesis Data Analytics console, choose the arrow next to your application, and then choose Application details.

3. Choose Connect to a destination.

4. Choose the Kinesis Firehose stream you created in step 1, leave the rest of the options at their default settings, and then choose Save and continue.

In several seconds, you return to the main page of the application.
5. Choose Go to SQL results.

6. Choose Save and run SQL to re-run the query you ran in Step 5: Run a SQL Query (p. 321).

Amazon Kinesis Data Analytics attempts to process event data it receives from the Kinesis Firehose delivery stream. If you encounter the No rows have arrived yet error, ensure that you are still sending emails so that Amazon Kinesis Data Analytics has email sending events to process.

As Amazon Kinesis Data Analytics processes records, results appear in the Real-time analytics tab. Amazon Kinesis Data Analytics automatically saves the results to the Amazon S3 bucket that you specified when you set up the Kinesis Firehose delivery stream in step 1.
7. To retrieve the results, go to the Amazon S3 console.
8. Choose the Amazon S3 bucket that is associated with the Kinesis Firehose delivery stream that the Amazon Kinesis Data Analytics application uses as its destination.
9. Navigate to the data, which, by default, is organized in a folder hierarchy based on the date the results are saved to the bucket.

   If the bucket is empty, wait a few minutes and try again. It can take several minutes for data to get from Amazon Kinesis Data Analytics to your Amazon S3 bucket.

10. Choose a file, and then from the Actions menu, choose Download.

11. Follow the on-screen instructions to download the file to your computer.

12. On your computer, open the file with a text editor. The records are in JSON format, and each record is contained in curly braces. The following is an example of a file that contains two records.
Configuring Custom Domains to Handle Open and Click Tracking

When you use event publishing (p. 247) to capture open and click events, Amazon SES makes minor changes to the emails you send. To capture open events, Amazon SES adds a 1 pixel by 1 pixel transparent image to the bottom of each email. This image has a unique file name for each email, and is hosted on a server operated by Amazon SES. To capture link click events, Amazon SES replaces the links in your emails with links to a server operated by Amazon SES. This immediately redirects the recipient to his or her intended destination. Some Amazon SES customers may want to use their own domains, rather than domains owned and operated by Amazon SES, to create a more cohesive experience for their recipients.

You can configure multiple custom domains to handle open and click tracking events. These custom domains are associated with configuration sets. When you send an email using a configuration set, if that configuration set is configured to use a custom domain, then the open and click links in that email automatically use the custom domain specified in that configuration set.

This section contains procedures for setting up a subdomain on a server you own to automatically redirect users to the open and click tracking servers operated by Amazon SES. There are two steps involved in setting up these domains. First, you configure the subdomain itself, and then you set up a configuration set to use the custom domain. This topic contains procedures for completing both of these steps.

Part 1: Setting up a Domain for Handling Open and Click Link Redirects

The specific procedures for setting up a redirect domain vary depending on your web hosting provider (and your Content Delivery Network, if you use an HTTPS server). The procedures in the following sections provide general guidance rather than specific steps.

Option 1: Configuring an HTTP Domain

If the subdomain you will use for handling open and click links uses HTTP rather than HTTPS, the process for configuring the subdomain involves only a few steps.

Note
If you set up a custom domain that uses the HTTP protocol, and you send an email that contains links that use the HTTPS protocol, your customers may see a warning message when they click the links in your email. If you plan to send emails that contain links that use the HTTPS protocol, you should use an HTTPS domain for handling open and click tracking events.

If you plan to use an HTTPS subdomain, follow the procedures in Option 2: Configuring an HTTPS Domain (p. 327) instead.
To set up an HTTP subdomain for handling open and click links

1. If you have not already done so, create a subdomain to use for open and click tracking links. We recommend that you create a subdomain that is dedicated specifically to handling these links.
2. Verify the subdomain you created in step 1 for use with Amazon SES. For more information, see Verifying Domains in Amazon SES (p. 52).
3. Modify the DNS record for the subdomain you created in step 1. In the DNS record, add a new CNAME record to refer to the appropriate tracking domain for the AWS Region in which you are using Amazon SES. The following table contains a list of tracking domains for the AWS Regions in which Amazon SES is available.

<table>
<thead>
<tr>
<th>AWS Region</th>
<th>AWS tracking domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>US West (Oregon)</td>
<td>r.us-west-2.awstrack.me</td>
</tr>
<tr>
<td>US East (N. Virginia)</td>
<td>r.us-east-1.awstrack.me</td>
</tr>
<tr>
<td>EU (Ireland)</td>
<td>r.eu-west-1.awstrack.me</td>
</tr>
</tbody>
</table>

**Note**
Depending on your web hosting provider, it may take several minutes for the changes you make to the subdomain's DNS record to take effect. Your web hosting provider or IT organization can provide additional information about these delays.

Option 2: Configuring an HTTPS Domain

If you plan to use an HTTPS subdomain for handling open and click links, you must perform some additional steps, beyond those required for setting up an HTTP domain (p. 326).

To set up an HTTPS subdomain for handling open and click links

1. Create a subdomain to use for open and click tracking links. We recommend that you create a subdomain that is dedicated specifically to handling these links.
2. Verify the subdomain you created in step 1 for use with Amazon SES. For more information, see Verifying Domains in Amazon SES (p. 52).
3. Create a new account with a Content Delivery Network (CDN), such as Amazon CloudFront.
4. Configure the CDN to redirect requests to the appropriate tracking domain for the AWS Region in which you are using Amazon SES. The following table contains a list of tracking domains for the AWS Regions in which Amazon SES is available.

<table>
<thead>
<tr>
<th>AWS Region</th>
<th>AWS tracking domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>US West (Oregon)</td>
<td>r.us-west-2.awstrack.me</td>
</tr>
<tr>
<td>US East (N. Virginia)</td>
<td>r.us-east-1.awstrack.me</td>
</tr>
<tr>
<td>EU (Ireland)</td>
<td>r.eu-west-1.awstrack.me</td>
</tr>
</tbody>
</table>

5. If you use Amazon CloudFront as your CDN, complete the following procedures:
   a. On the CloudFront Distributions page, choose the distribution that corresponds with your CDN.
   b. On the Behaviors tab, choose the default behavior, and then choose Edit.
   c. For Cache Based on Selected Request Headers, choose All.
d. For **Query String Forwarding and Caching**, choose **Forward all, cache based on all**.

If you use a CDN other than CloudFront, you may need to complete similar steps. See your CDN's documentation for more information.

6. Modify the DNS record for the subdomain you created in step 1. In the DNS record, add a new CNAME record to refer to your address in the CDN you set up in step 3.

   **Note**
   Depending on your web hosting provider, it may take several minutes for the changes you make to the subdomain's DNS record to take effect. Your web hosting provider or IT organization can provide additional information about these delays.

7. Acquire an SSL certificate from a trusted Certificate Authority. The certificate should cover your address at the CDN, as well as the subdomain you created in step 1. Upload the certificate to the CDN.

### Part 2: Setting up a Configuration Set to Refer to a Custom Open and Click Tracking Domain

After you configure your domain to handle open and click tracking redirects, you must set up an event destination in a configuration set to refer to your custom domain. You can complete this step using the Amazon SES console or the CreateConfigurationSetTrackingOptions API operation. This section contains procedures for completing these tasks using the Amazon SES console; for more information about using the API, see CreateConfigurationSetTrackingOptions in the *Amazon SES API Reference*.

**To create a new configuration set event destination that refers to a custom tracking domain**

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation bar on the left side of the screen, choose **Configuration Sets**.
3. Choose **Create Configuration Set**.
4. For **Configuration Set Name**, type a name for the configuration set, and then choose **Create Configuration Set**.
5. In the list of configuration sets, select the box next to the configuration set you created in the previous step. On the **Actions** menu, choose **Edit**.
6. On the **Event Destinations** tab, for **Add Destination**, choose an event destination type. For more information about the options in this menu, see Add an Event Destination (p. 249).
7. For **Event types**, choose either **Click**, **Open**, or both, depending on the types of events you want to track.
8. For **Domain**, choose **Use your own subdomain**.
9. For **Select a verified domain**, choose the domain that you want to use for open and click event tracking. In the text field to the left of the menu, you can optionally specify a subdomain of the parent domain.
10. Configure the remaining options as you normally would. For more information about setting up event destinations, see Add an Event Destination (p. 249).
11. Choose **Save**.
Monitoring Your Amazon SES Sender Reputation

Amazon SES actively tracks several metrics that may cause your reputation as a sender to be damaged, or that could cause your email delivery rates to decline. Two important metrics that we consider in this process are the bounce and complaint rates for your account. Accounts with excessively high bounce or complaint rates may be placed on probation or suspended. For more information, see Amazon SES Enforcement FAQs (p. 436).

Because your bounce and complaint rate are so important to the health of your account, Amazon SES includes a reputation dashboard that you can use to track these metrics. The reputation dashboard can also display information about factors unrelated to bounces or complaints that could damage your sender reputation. For example, if you send email to a known spamtrap, you will see a message on this dashboard.

This section contains information about accessing the reputation dashboard, interpreting the information it contains, and setting up systems to actively notify you of factors that could impact your sender reputation.

In this section, you will find the following topics:

- Using the Reputation Dashboard to Track Bounce and Complaint Rates (p. 329)
- Reputation Dashboard Messages (p. 330)
- Creating Reputation Monitoring Alarms Using CloudWatch (p. 341)
- Automatically Pausing Email Sending (p. 343)

Using the Reputation Dashboard to Track Bounce and Complaint Rates

The reputation dashboard contains the same information that the Amazon SES team sees when determining the health of individual accounts.

To view the reputation dashboard

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation pane on the left side of the screen, choose Reputation Dashboard.

The dashboard displays the following information:

- **Account status** – A brief description of the health of your account. Possible values include:
  - **Healthy** – There are no issues currently impacting your account.
  - **Probation** – Your account is on probation. If the issues that caused your account to be put on probation are not resolved by the end of the probation period, your account may be suspended.
- **Pending end of probation decision** – Your account is on probation. Because of the nature of the issues that led to this probation, a member of the Amazon SES team must review your account before any further action is taken.

- **Shutdown** – Your account has been shutdown. While your account is shutdown, you will not be able to send email using Amazon SES. You can appeal this decision; see *Amazon SES Enforcement FAQs* (p. 436) for more information about submitting an appeal.

- **Pending shutdown** – Your account is on probation. The issues that caused this probation have not been resolved. These issues may lead to your account being shut down. However, because of the nature of your account, a member of the Amazon SES team will review your account before any further action is taken.

- **Bounce Rate** – The percentage of emails sent from your account that resulted in a hard bounce.

- **Complaint Rate** – The percentage of emails sent from your account that resulted in recipients reporting them as spam.

  **Note**
  The Bounce Rate and Complaint Rate sections also include status messages for their respective metrics. The following is a list of status messages that may be displayed for these metrics:

  - **Healthy** – The metric is below the threshold that could cause your account to be placed on probation.

  - **Almost healed** – The metric caused your account to be placed on probation. Since the probation began, the metric has stayed below the maximum rate. If the metric remains below the maximum rate, its status may change to Healthy before the probation period ends.

  - **Probation** – The metric caused your account to be placed on probation, and is still above the maximum rate. If the issue that caused the metric to exceed the maximum rate is not resolved by the end of the probation period, your account may be suspended.

  - **Shutdown** – The metric caused your account to be suspended. While your account is shut down, you cannot send email using Amazon SES. You can appeal this decision. For information about submitting an appeal, see *Amazon SES Enforcement FAQs* (p. 436).

  - **Pending shutdown** – The metric caused your account to be placed on probation. The issues that caused this probation have not been resolved. These issues may lead to your account being shut down. A member of the Amazon SES team will review your account before any further action is taken.

  - **Other Notifications** – If your account is experiencing reputation-related issues that are not related to bounces or complaints, a brief message will be shown here. For more information about the notifications that can be shown in this area, see *Reputation Dashboard Messages* (p. 330).

---

**Reputation Dashboard Messages**

The Amazon SES reputation dashboard provides important metrics related to your account. The following sections describe the messages that may be displayed in this dashboard, and provide tips and information that you may be able to use to resolve issues related to your sender reputation.

This section contains information about the following types of notifications:

- **Status Messages** (p. 331)
- **Bounce Rate Notification** (p. 331)
- **Complaint Rate Notification** (p. 332)
- **Anti-Spam Organization Notification** (p. 333)
- **Direct Feedback Notification** (p. 334)
• Domain Blocklist Notification (p. 335)
• Internal Review Notification (p. 336)
• Mailbox Provider Notification (p. 337)
• Recipient Feedback Notification (p. 338)
• Related Account Notification (p. 339)
• Spamtrap Notification (p. 339)
• Vulnerable Site Notification (p. 340)
• Other Notification (p. 341)

Status Messages

When you use the reputation dashboard, you will see a message describing the status of your Amazon SES account. The following is a list of possible account status values:

• Healthy – There are no issues currently impacting your account.
• Probation – Your account is on probation. If the issues that caused your account to be put on probation are not resolved by the end of the probation period, your account may be suspended.
• Pending end of probation decision – Your account is on probation. Because of the nature of the issues that led to this probation, a member of the Amazon SES team must review your account before any further action is taken.
• Shutdown – Your account has been shutdown. While your account is shutdown, you will not be able to send email using Amazon SES. You can appeal this decision; see Amazon SES Enforcement FAQs (p. 436) for more information about submitting an appeal.
• Pending shutdown – Your account is on probation. The issues that caused this probation have not been resolved. These issues may lead to your account being shut down. However, because of the nature of your account, a member of the Amazon SES team will review your account before any further action is taken.

Additionally, the Bounce Rate and Complaint Rate sections of the reputation dashboard display status summaries for their respective metrics. The following is a list of possible metric status values:

• Healthy – The metric is below the threshold that could cause your account to be placed on probation.
• Almost healed – The metric caused your account to be placed on probation. Since the probation began, the metric has stayed below the maximum rate. If the metric remains below the maximum rate, its status may change to Healthy before the probation period ends.
• Probation – The metric caused your account to be placed on probation, and is still above the maximum rate. If the issue that caused the metric to exceed the maximum rate is not resolved by the end of the probation period, your account may be suspended.
• Shutdown – The metric caused your account to be suspended. While your account is shut down, you cannot send email using Amazon SES. You can appeal this decision. For information about submitting an appeal, see Amazon SES Enforcement FAQs (p. 436).
• Pending shutdown – The metric caused your account to be placed on probation. The issues that caused this probation have not been resolved. These issues may lead to your account being shut down. A member of the Amazon SES team will review your account before any further action is taken.

Bounce Rate Notification

This section contains additional information about bounce rate notifications shown in the Amazon SES reputation dashboard.
**Why you received this notification**

Amazon SES recommends that senders maintain bounce rates below 5%; accounts with bounce rates exceeding 10% will be placed on probation, and may be at risk for suspension.

The bounce rate is based on the number of hard bounces generated by your Amazon SES account. A hard bounce occurs when an email is sent to an address that does not exist. Soft bounces, which occur when a recipient's address is temporarily unable to receive your message, are not considered in this calculation. Bounced emails sent to verified addresses and domains, as well as emails sent to the Amazon SES inbox simulator, are also not considered in this calculation.

Most email providers interpret a high bounce rate as a sign that a sender is not properly managing their recipient list, and that the sender may be sending unsolicited email. Because email providers take bounce rates seriously, Amazon SES takes swift action to protect your sender reputation.

**What you can do to resolve the issue**

If you have not done so already, put a process in place to capture and manage bounces and complaints. All Amazon SES accounts are required to have these processes in place. For more information, see [Email Program Success Metrics](p. 416).

Next, determine which email addresses are bouncing, and create and implement a plan for reducing or eliminating these bounces. If your account has already been shut down, contact us at the email address shown in the email you received from the Amazon SES Enforcement team to request a list of recent complaints.

**If your account is on probation**

If, at the end of your probation period, the bounce rate for your account remains above 10%, we will suspend your Amazon SES account.

If you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. In your email, describe the changes you implemented. If we agree that the changes will reduce your bounce rate, we will adjust our calculations to only consider bounces received after your changes were implemented.

**If your account is suspended**

You can appeal this suspension; see the [Enforcement FAQ](p. 439) for more information.

When you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. Include details of the actions you have taken to resolve this issue, as well as details of your plans to ensure that this issue does not occur again. We will consider your appeal and change the status of your account if necessary.

**Complaint Rate Notification**

This section contains additional information about complaint rate notifications shown in the Amazon SES reputation dashboard.

**Why you received this notification**

Amazon SES recommends that senders maintain complaint rates below 0.1%; accounts with complaint rates exceeding 0.5% will be placed on probation, and may be at risk for suspension.
The complaint rate is based on the number of complaint reports that are generated by the emails sent from your Amazon SES account. Not all email providers supply Amazon SES with information about the numbers of complaints they receive, so this value is adjusted based on the number of providers who do supply this information.

Most email providers interpret a high complaint rate as a sign that a sender is sending unwanted or unsolicited email. Because email providers take complaint rates seriously, Amazon SES takes swift action to protect your sender reputation.

What you can do to resolve the issue

If you have not done so already, put a process in place to capture and manage bounces and complaints. All Amazon SES accounts are required to have these processes in place. For more information, see Email Program Success Metrics (p. 416).

Next, determine which messages you are sending that result in complaints, and implement a plan for reducing these complaints. If your account has already been shut down, contact us at the address shown in the email you received from the Amazon SES Enforcement team to request a list of recent complaints.

While you should immediately stop sending to addresses that have complained, it is important that you identify the factors that are causing recipients to issue complaints. After you identify these factors, adjust your email sending behaviors to address them.

If your account is on probation

If, at the end of your probation period, the complaint rate for your account remains above 0.5%, we will suspend your Amazon SES account.

If you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. In your email, describe the changes you implemented. If we agree that the changes will reduce your complaint rate, we will adjust our calculations to only consider complaints received after your changes were implemented.

If your account is suspended

You can appeal this suspension; see the Enforcement FAQ (p. 439) for more information.

When you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. Include details of the actions you have taken to resolve this issue, as well as details of your plans to ensure that this issue does not occur again. We will consider your appeal and change the status of your account if necessary.

Anti-Spam Organization Notification

This section contains additional information about anti-spam organization notifications shown in the Amazon SES reputation dashboard.

Why you received this notification

A reputable anti-spam organization has reported that some of the content being sent from your Amazon SES account has been flagged as unsolicited or problematic by their systems.

Amazon SES does not have detailed information related to the specific messages that caused the anti-spam organization to flag your content as problematic. We cannot provide the name of the organization that issued the report. Typically, anti-spam organizations consider a combination of the following factors: recipient feedback, message engagement metrics, attempted deliveries to invalid addresses,
content that is flagged by their spam filters, and spamtrap hits. This is not an exhaustive list; other factors may cause these organizations to flag your content.

**What you can do to resolve the issue**

To resolve this issue, you need to determine what aspects of your email sending program may be causing the anti-spam organization to flag your email as problematic. You then need to change your sending program to address those issues.

**If your account is on probation**

If, at the end of your probation period, the anti-spam organization is still flagging your messages as problematic, we will suspend your Amazon SES account.

If you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. When we receive this information, we will extend the probation period to ensure that we are only analyzing the anti-spam organization notifications we have received after you implemented your changes. If, at the end of this extended probation period, your account is no longer listed by the anti-spam organization, we may remove the probation from your account.

**If your account is suspended**

You can appeal this suspension; see the Enforcement FAQ (p. 439) for more information.

When you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. Include details of the actions you have taken to resolve this issue, as well as details of your plans to ensure that this issue does not occur again. We will consider your appeal and change the status of your account if necessary.

**Direct Feedback Notification**

This section contains additional information about direct feedback notifications shown in the Amazon SES reputation dashboard.

**Why you received this notification**

A significant number of users have contacted Amazon SES directly to report messages that they received from an address or domain associated with your Amazon SES account. This type of feedback is not visible in the complaints reported by mailbox providers directly, and is not included in the bounce and complaint metrics shown on the reputation dashboard.

To protect the privacy of the users who reported these issues, we cannot provide their email addresses.

Recipients may complain to Amazon SES when they receive messages that they did not sign up to receive, when they don't receive the type of mail they expected to receive, when they do not find the email they receive to be useful or interesting, when they don't recognize that the messages are something that they signed up for, or when they are receiving too many messages. This list is not exhaustive; the factors that are relevant in your case depend on your specific email sending program.

**What you can do to resolve the issue**

We recommend that you implement a double opt-in strategy, as described in Building and Maintaining Your Lists (p. 419), for acquiring new addresses, and that you only send email to addresses that complete the double opt-in process.
Additionally, you should purge your lists of addresses that have not interacted with your emails recently. You can use open and click tracking, as described in Monitoring Your Amazon SES Sending Activity (p. 221), to determine which users are viewing and interacting with the content you send.

If your account is on probation

If, at the end of your probation period, Amazon SES is still receiving a significant volume of direct complaints about messages sent from your account, we will suspend your Amazon SES account.

If you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. When we receive this information, we will extend the probation period to ensure that we are only analyzing the amount of direct feedback we received after you implemented your changes. At the end of this extended probation period, if the amount of direct feedback has been reduced or eliminated, we may remove the probation from your account.

If your account is suspended

You can appeal this suspension; see the Enforcement FAQ (p. 439) for more information.

When you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. Include details of the actions you have taken to resolve this issue, as well as details of your plans to ensure that this issue does not occur again. We will consider your appeal and change the status of your account if necessary.

Domain Blocklist Notification

This section contains additional information about domain blocklist notifications shown in the Amazon SES reputation dashboard.

Why you received this notification

Emails sent from your Amazon SES account contain references to domains that have been listed on a reputable Domain Blocklist. Domains on these lists are typically associated with abusive or malicious behavior. The domains in question may or may not be the domains from which you are sending email. Messages that include references or links to a domain on a blocklist, or that include images hosted on such a domain, may also be flagged.

Amazon SES is unable to provide the names of the domains that are causing your message to be flagged, or to identify which emails were flagged in this way.

What you can do to resolve the issue

Create a list of all of the domains referenced in the emails you send through Amazon SES, and then use the Spamhaus Domain Lookup tool at https://www.spamhaus.org/lookup/ to determine which domains referenced in your email are on the domain blocklist. More than one domain referenced in the emails you send may be on this blocklist.

Amazon SES is not affiliated with the Spamhaus Domain Blocklist, and makes no guarantees about the accuracy of the domains on this list. The Spamhaus Domain Blocklist and Domain Lookup Tool are owned, operated, and maintained by the Spamhaus Project.

If your account is on probation

If, at the end of your probation period, a significant portion of the email you send through Amazon SES still contains references to domains on the block list, we will suspend your Amazon SES account.
If you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. When we receive this information, we will extend the probation period to ensure that we are only analyzing the number of blocklisted domains present in your email after you put your changes in place. At the end of this extended probation period, if the amount of domain blocklist notifications has been reduced or eliminated, we may remove the probation from your account.

If your account is suspended

You can appeal this suspension; see the Enforcement FAQ (p. 439) for more information.

When you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. Include details of the actions you have taken to resolve this issue, as well as details of your plans to ensure that this issue does not occur again. We will consider your appeal and change the status of your account if necessary.

Internal Review Notification

This section contains additional information about internal review notifications shown in the Amazon SES reputation dashboard.

Why you received this notification

A comprehensive review of your account identified several characteristics that may cause mailbox providers or recipients to identify your messages as spam.

To protect our abuse detection process, we cannot reveal the specific factors that led to your account being flagged in this way.

Common factors that can lead to this determination include the following:

- Messages being flagged by commercial anti-spam systems.
- Message content that implies the recipient has not explicitly requested the email.
- Mismatches between the message sender and the branding within the email body.
- Content that does not make it obvious who the sender is.
- Sending messages that deal with content that is associated with unsolicited email.
- Formatting patterns associated with unsolicited email.
- Sending from or making reference to domains with poor reputations.

This is not a comprehensive list; the specific reason for this notification may be a combination of any of these factors, or the reason might be something not listed.

What you can do to resolve the issue

The following suggestions may help reduce the severity of the issue:

- Ensure that the only recipients you are contacting are those who have explicitly asked to receive email from you.
- Never purchase, rent, or borrow lists of email recipients.
- Do not attempt to hide your identity or the purpose of your communication in the messages you send.
- Create a list of all of the domains referenced in the emails you send through Amazon SES, and then use the Spamhaus Domain Lookup tool at https://www.spamhaus.org/lookup/ to determine if any of those domains are on the Spamhaus Domain Blocklist.
- Ensure that you are following industry best practices when designing your emails.

This list is not exhaustive, but it should help you identify some of the most common factors that may lead to your email being flagged.

Amazon SES is not affiliated with the Spamhaus Domain Blocklist, and makes no guarantees about the accuracy of the domains on this list. The Spamhaus Domain Blocklist and Domain Lookup Tool are owned, operated, and maintained by the Spamhaus Project.

**If your account is on probation or suspended**

When you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. Include details of the actions you have taken to resolve this issue, as well as details of your plans to ensure that this issue does not occur again. We will consider your appeal and change the status of your account if necessary.

If, after your probation or shutdown is listed, we detect the issue again, your account may be subject to an additional probation or shutdown. Repeated incidents may lead to your Amazon SES account being permanently disabled.

See Amazon SES Enforcement FAQs (p. 436) for more information about probation and suspension of Amazon SES accounts.

**Mailbox Provider Notification**

This section contains additional information about mailbox provider notifications shown in the Amazon SES reputation dashboard.

**Why you received this notification**

A major mailbox provider has reported to us that unsolicited or malicious email is being sent from an address or domain associated with your Amazon SES account.

We cannot share the identity of the organization that issued this report. Additionally, we do not have information about the specific factors that caused the mailbox provider to issue the report. Typically, mailbox providers make this kind of determination based on customer feedback, customer engagement metrics, attempted deliveries to invalid addresses, and content that is flagged by spam filters. This list is not exhaustive; there may be other factors that caused the mailbox provider to flag your content.

**What you can do to resolve the issue**

To resolve this issue, you need to determine which aspects of your email sending program may have caused mailbox providers to flag your mail as being problematic. You must then change your sending program to address those issues.

**If your account is on probation**

If, at the end of your probation period, the mailbox provider is still flagging your email as being problematic, we may suspend your Amazon SES account.

If you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. When we receive this information, we will extend the probation period to ensure that we are only analyzing the number of mailbox provider notifications we receive after you implement your changes. At the end of this extended
probation period, if the mailbox provider no longer reports your account as being problematic, we may remove the probation from your account.

If your account is suspended

You can appeal this suspension; see the Enforcement FAQ (p. 439) for more information.

When you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. Include details of the actions you have taken to resolve this issue, as well as details of your plans to ensure that this issue does not occur again. We will consider your appeal and change the status of your account if necessary.

Recipient Feedback Notification

This section contains additional information about recipient feedback notifications shown in the Amazon SES reputation dashboard.

Why you received this notification

A major mailbox provider has reported to us that large numbers of their users are reporting mail sent from your Amazon SES account as unsolicited. This type of feedback is not visible in the complaints reported by mailbox providers directly, and is not included in the Amazon SES bounce and complaint notifications.

A large number of complaints can have a negative impact on all Amazon SES users. To protect your reputation and that of other Amazon SES customers, we take immediate action when an account receives a certain number of complaints.

We are unable to provide a list of the specific email addresses that are reporting your email as unsolicited. Additionally, we are unable to share the name of the mailbox provider that has reported this issue to us.

What you can do to resolve the issue

To resolve this issue, you need to determine which aspects of your email sending program may be causing your recipients to issue complaints against the email messages they receive from you. After you identify these factors, change your email sending practices to correct them.

To acquire new addresses, we recommend that you implement a double opt-in strategy, as described in Building and Maintaining Your Lists (p. 419). We recommend that you only send email to addresses that have completed the double opt-in process.

Additionally, you should purge your lists of addresses that have not interacted with your emails recently. You can use open and click tracking, as described in Monitoring Your Amazon SES Sending Activity (p. 221), to determine which users are viewing and interacting with the content you send.

If your account is on probation

If, at the end of your probation period, the mailbox provider still reports a significant number of complaints, we will suspend your Amazon SES account.

If you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. When we receive this information, we will extend the probation period to ensure that we are only analyzing the number of mailbox provider complaints that we receive after you implement your changes. At the end of
this extended probation period, if the number of mailbox provider complaints has been reduced or eliminated, we may remove the probation from your account.

If your account is suspended

You can appeal this suspension; see the Enforcement FAQ (p. 439) for more information.

When you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. Include details of the actions you have taken to resolve this issue, as well as details of your plans to ensure that this issue does not occur again. We will consider your appeal and change the status of your account if necessary.

Related Account Notification

This section contains additional information about related account notifications shown in the Amazon SES reputation dashboard.

Why you received this notification

We have detected serious problems related to emails sent from another Amazon SES account. We believe that the problematic account is related to your AWS account, so we have taken action to avoid similar problems.

What you can do to resolve the issue

The related account should have received an email describing the issues that caused it to be suspended. Address the issues with the related account first.

When you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. Include details of the actions you have taken to resolve this issue, as well as details of your plans to ensure that this issue does not occur again. If we agree with your assessment, we may lift the probation or suspension on your account.

Spamtrap Notification

This section contains additional information about spamtrap notifications shown in the Amazon SES reputation dashboard.

Why you received this notification

A third-party anti-spam organization has reported to us that their spamtrap addresses recently received email from a verified address or domains associated with your Amazon SES account.

A spamtrap is a dormant email address that is used exclusively to lure unsolicited email (spam). A large number of spamtrap reports can have a negative impact on all Amazon SES users. To protect your reputation and that of other Amazon SES customers, we take immediate action when an account sends a particular volume of email to spamtrap addresses.

What you can do to resolve the issue

We cannot reveal the email addresses associated with the spamtrap you encountered. These addresses are closely guarded by the organizations that own them, and once the addresses are known, they become worthless.
Sending email to spamtrap addresses typically indicates that there is an issue with how you acquire your customers’ email addresses. For example, purchased lists of email addresses can contain spamtrap addresses, which is why sending to purchased or rented lists is prohibited by the Amazon SES terms of service. To acquire new addresses, we recommend that you implement a double opt-in strategy, as described in Building and Maintaining Your Lists (p. 419). We recommend that you only send email to addresses that have completed the double opt-in process.

Additionally, you should purge your lists of addresses that have not interacted with your emails recently. You can use open and click tracking, as described in Monitoring Your Amazon SES Sending Activity (p. 221), to determine which users are viewing and interacting with the content you send.

**If your account is on probation**

If, at the end of your probation period, messages sent from your account are still being sent to spamtrap addresses, we will suspend your Amazon SES account.

If you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. When we receive this information, we will extend the probation period to ensure that we are only analyzing the number of spamtrap reports we receive after you implement your changes. At the end of this extended probation period, if the number of spamtrap reports has been reduced or eliminated, we may remove the probation from your account.

**If your account is suspended**

You can appeal this suspension; see the Enforcement FAQ (p. 439) for more information.

When you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. Include details of the actions you have taken to resolve this issue, as well as details of your plans to ensure that this issue does not occur again. We will consider your appeal and change the status of your account if necessary.

**Vulnerable Site Notification**

This section contains additional information about vulnerable site notifications shown in the Amazon SES reputation dashboard.

**Why you received this notification**

A comprehensive review has found that messages are being sent from your account that we do not believe you intended to send. These messages are highly likely to be flagged as spam by mailbox providers and recipients.

Most often in these situations, a third party is abusing a feature of your website to send unwanted email. For example, if your website contains an "email to a friend," "contact us," "invite a friend," or similar feature, a third party can use that feature to send unsolicited email.

**What you can do to resolve the issue**

First, identify features of your website or applications that may allow third parties to send emails using Amazon SES without your knowledge. Contact us at the email address shown in the email you received from the Amazon SES Enforcement team to receive a sample of the email messages we believe were sent in this manner.

Next, modify your application or website to prevent unsolicited sending. For example, add a CAPTCHA, limit the rate at which emails can be sent, remove the ability of users to submit custom content,
require users to log in to send email, and remove the ability for the application to generate multiple simultaneous notifications.

**If your account is on probation or suspended**

When you have implemented changes that you believe will resolve the issue, please email us at the address shown in the email you received from the Amazon SES Enforcement team. Include details of the actions you have taken to resolve this issue, as well as details of your plans to ensure that this issue does not occur again. We will consider your appeal and change the status of your account if necessary.

If, after your probation or shutdown is listed, we detect the issue again, your account may be subject to an additional probation or shutdown. Repeated incidents may lead to your Amazon SES account being permanently disabled.

See Amazon SES Enforcement FAQs (p. 436) for more information about probation and suspension of Amazon SES accounts.

**Other Notification**

This section contains additional information about other notifications shown in the Amazon SES reputation dashboard.

**Why you received this notification**

An automatic or human review has identified issues that are not listed in the previous sections of this document.

**What you can do to resolve the issue**

Refer to the email you received from the Amazon SES Enforcement team for details on the specific issue. If possible, address this issue and reply to the ticket with the steps you have taken to address the issue. Depending on your specific situation and the nature of the issues we discovered, we may lift your probation or suspension.

---

**Creating Reputation Monitoring Alarms Using CloudWatch**

Amazon SES automatically publishes two reputation-related metrics to Amazon CloudWatch: Reputation.BounceRate and Reputation.ComplaintRate. You can use these metrics to create alarms that notify you when your bounce or complaint rates reach levels that could impact your ability to send email.

The values of Reputation.BounceRate and Reputation.ComplaintRate are decimal numbers between 0 and 1. Multiply either of these metrics by 100 to determine the percentage of your emails that are bouncing or resulting in complaints.

**To create a CloudWatch alarm**

1. Sign in to the AWS Management Console and open the Amazon SNS console at https://console.aws.amazon.com/sns/v2/home.
2. Create a new Amazon SNS topic, and then subscribe to it using your preferred endpoint (such as email or SMS). For more information, see Create a Topic and Subscribe to a Topic in the Amazon Simple Notification Service Developer Guide.
4. In the navigation panel on the left side of the screen, choose Alarms.
5. Choose Create Alarm.
6. Under CloudWatch Metrics by Category, choose SES Metrics.
7. Under Metric name, choose the metric for which you want to create the alarm.

For example, if you want to create an alarm when your bounce rate reaches a certain level, choose Reputation.BounceRate. If you want to create an alarm when your complaint rate reaches a certain level, choose Reputation.ComplaintRate.

Choose Next.
8. Under Alarm Threshold, for Name, type a name for the alarm.
9. In the alarm threshold area, specify the value that should cause CloudWatch to raise an alarm.

If you are creating an alarm for Reputation.BounceRate, note that Amazon SES recommends that you maintain a bounce rate under 5%, and that a rate of 10% puts your account at risk for suspension. For this reason, we recommend that you configure CloudWatch to raise an alarm when the bounce rate is greater than or equal to 0.05 (5%), as shown in the following image.

![Whenever: Reputation.BounceRate
is: >= 0.05
for: 1 consecutive period(s)](image)

If you are creating an alarm for Reputation.ComplaintRate, note that Amazon SES recommends that you maintain a complaint rate under 0.1%, and that a rate of 0.5% puts your account at risk for suspension. For this reason, we recommend that you configure CloudWatch to raise an alarm when the bounce rate is greater than or equal to 0.001 (0.1%), as shown in the following image.

![Whenever: Reputation.ComplaintRate
is: >= 0.001
for: 1 consecutive period(s)](image)

10. Under Additional settings, for Treat missing data as, choose ignore (maintain the alarm state).
11. Under Alarm Preview, for Period, choose an alarm period.

Note
Amazon SES sends Reputation.BounceRate and Reputation.ComplaintRate metrics to CloudWatch every 15 minutes. For this reason, we recommend that you choose an alarm period of 15 minutes or greater. You will not see any added benefit by choosing an alarm period of less than 15 minutes.

12. Under Actions, for Whenever this alarm, choose State is ALARM. For Send notification to, choose the topic that you created and subscribed to in step 2.
13. Choose Create Alarm.
Automatically Pausing Email Sending

To protect your sender reputation, you can temporarily pause email sending for messages sent using specific configuration sets, or for all messages sent from your Amazon SES account in a specific AWS Region.

By using Amazon CloudWatch and Lambda, you can create a solution that automatically pauses your email sending when your reputation metrics (such as bounce rate or complaint rate) exceed certain thresholds. This topic contains procedures for setting up this solution.

Topics in this section:
• Automatically Pausing Email Sending for Your Amazon SES Account (p. 343)
• Automatically Pausing Email Sending for a Configuration Set (p. 347)

Automatically Pausing Email Sending for Your Amazon SES Account

The procedures in this section explain the steps to set up Amazon SES, Amazon SNS, Amazon CloudWatch, and AWS Lambda to automatically pause email sending for your Amazon SES account in a single AWS Region. If you send email from multiple regions, repeat the procedures in this section for each region in which you want to implement this solution.

Topics in this section:
• Part 1: Create an IAM Role (p. 343)
• Part 2: Create the Lambda Function (p. 344)
• Part 3: Re-Enable Email Sending for Your Account (p. 345)
• Part 4: Create an Amazon SNS Topic (p. 345)
• Part 5: Create a CloudWatch Alarm (p. 346)
• Part 6: Test the solution (p. 347)

Part 1: Create an IAM Role

The first step in configuring automatic pausing of email sending is to create an IAM role that can execute the UpdateAccountSendingEnabled API operation.

To create the IAM role
1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose Roles.
3. Choose Create role.
4. Under Select type of trusted entity, choose AWS service.
5. Under Choose the service that will use this role, choose Lambda. Choose Next: Permissions.
6. On the Attach permissions policies page, choose the following policies:
   • AWSLambdaBasicExecutionRole
Part 2: Create the Lambda Function

After you create an IAM role, you can create the Lambda function that pauses email sending for your account.

To create the Lambda function

1. Open the AWS Lambda console at https://console.aws.amazon.com/lambda/.
2. Use the region selector to choose the region in which you want to deploy this Lambda function.
   
   **Note**
   This function only pauses email sending in the AWS Region you select in this step. If you send email from more than one region, repeat the procedures in this section for each region in which you want to automatically pause email sending.

3. Choose **Create function**.
4. Under **Create function**, choose **Author from scratch**.
5. Under **Author from scratch**, complete the following steps:
   
   - For **Name**, type a name for the Lambda function.
   - For **Runtime**, choose **Node.js 6.10**.
   - For **Role**, choose **Choose an existing role**.
   - For **Existing role**, choose the IAM role you created in the section called “Part 1: Create an IAM Role” (p. 343).

   Choose **Create function**.

6. Under **Function code**, in the code editor, paste the following code:

```javascript
'use strict';

var aws = require('aws-sdk');

// Create a new SES object.
var ses = new aws.SES();

// Specify the parameters for this operation. In this case, there is only one
// parameter to pass: the Enabled parameter, with a value of false
// (Enabled = false disables email sending, Enabled = true enables it).
var params = {
    Enabled: false
};

exports.handler = (event, context, callback) => {
    // Pause sending for your entire SES account
    ses.updateAccountSendingEnabled(params, function(err, data) {
        if(err) {
            console.log(err.message);
        } else {
```

---

Tip
Use the search box at the top of the list of policies to quickly locate these policies.
to complete these procedures, you must install and configure the AWS Command Line Interface. For more information, see the AWS Command Line Interface User Guide.

To re-enable email sending

1. At the command line, type the following command to re-enable email sending for your account: `aws ses update-account-sending-enabled --enabled --region us-west-2`

   **Note**
   Replace `us-west-2` in the preceding command with the name of the region in which you want to re-enable email sending.

2. At the command line, type the following command to check the email sending status for your account: `aws ses get-account-sending-enabled --region us-west-2`

   If you see the following output, then you have successfully re-enabled email sending for your account:

   ```json
   {
     "Enabled": true
   }
   ```

Part 4: Create an Amazon SNS Topic

For CloudWatch to execute your Lambda function when an alarm is triggered, you must first create an Amazon SNS topic and subscribe the Lambda function to it.

To create the Amazon SNS topic

2. Use the region selector to choose the region in which you want to automatically pause email sending.
3. In the navigation pane, choose Topics.
Part 5: Create a CloudWatch Alarm

This section contains procedures for creating an alarm in CloudWatch that is triggered when a metric reaches a certain threshold. When the alarm is triggered, it delivers a notification to the Amazon SNS topic you created in the section called “Part 4: Create an Amazon SNS Topic” (p. 345), which then executes the Lambda function you created in the section called “Part 2: Create the Lambda Function” (p. 344).

To create a CloudWatch alarm

2. Use the region selector to choose the region in which you want to automatically pause email sending.
3. In the navigation pane, choose Alarms.
4. Choose Create Alarm.
5. On the Create Alarm window, under SES Metrics, choose Account Metrics.
6. Under Metric Name, choose one of the following options:
   - Reputation.BounceRate – Choose this metric if you want to pause email sending for your account when the overall hard bounce rate for your account crosses a threshold that you define.
   - Reputation.ComplaintRate – Choose this metric if you want to pause email sending for your account when the overall complaint rate for your account crosses a threshold that you define.

   Choose Next.

7. Complete the following steps:
   - Under Alarm Threshold, for Name, type a name for the alarm.
   - Under Whenever: Reputation.BounceRate or Whenever: Reputation.ComplaintRate, specify the threshold that causes the alarm to trigger.

     Note
     
     Your account is automatically placed on probation if your bounce rate exceeds 10%, or if your complaint rate exceeds .5%. When you specify the bounce or complaint rate that causes the CloudWatch alarm to trigger, we recommend that you use values that are below these rates to prevent your account from being placed on probation.

   - Under Actions, for Whenever this alarm, choose State is ALARM. For Send notification to, choose the Amazon SNS topic you created in the section called “Part 4: Create an Amazon SNS Topic” (p. 345).
Choose **Create Alarm**.

### Part 6: Test the solution

You can now test the alarm to ensure that it executes the Lambda function when it enters the **ALARM** state. You can use the `SetAlarmState` API operation to temporarily change the state of the alarm.

The procedures in this section are optional, but we recommend that you complete them to ensure that the entire solution is configured correctly.

1. At the command line, type the following command to check the email sending status for your account:
   ```bash
   aws ses get-account-sending-enabled --region us-west-2
   ```
   **Note**
   Replace `us-west-2` in the preceding command with the name of the region you specified in the previous step.

   If sending is enabled for your account, you see the following output:
   ```json
   {   "Enabled": true
   }
   ```

2. At the command line, type the following command to temporarily change the alarm state to **ALARM**:
   ```bash
   aws cloudwatch set-alarm-state --alarm-name MyAlarm --state-value ALARM --state-reason "Testing execution of Lambda function" --region us-west-2
   ```
   Replace `MyAlarm` in the preceding command with the name of the alarm you created in the section called “Part 5: Create a CloudWatch Alarm” (p. 346), and replace `us-west-2` with the region in which you want to automatically pause email sending.
   **Note**
   When you execute this command, the status of the alarm switches from **OK** to **ALARM** and back to **OK** within a few seconds. You can view these status changes on the alarm’s **History** tab in the CloudWatch console, or by using the `DescribeAlarmHistory` operation.

3. At the command line, type the following command to check the email sending status for your account:
   ```bash
   aws ses get-account-sending-enabled --region us-west-2
   ```
   If the Lambda function executed successfully, you see the following output:
   ```json
   {   "Enabled": false
   }
   ```

4. Complete the steps in the section called “Part 3: Re-Enable Email Sending for Your Account” (p. 345) to re-enable email sending for your account.

### Automatically Pausing Email Sending for a Configuration Set

You can configure Amazon SES to export reputation metrics that are specific to emails that are sent using a specific configuration set to Amazon CloudWatch. You can then use these metrics to create CloudWatch alarms that are specific to these configuration sets. When these alarms exceed certain
thresholds, you can automatically pause the sending of emails that use the specified configuration sets, without impacting the overall email sending capabilities of your Amazon SES account.

**Note**
The solution described in this section pauses email sending for a specific configuration set in a single AWS Region. If you send email from multiple regions, repeat the procedures in this section for each region in which you want to implement this solution.

**Topics in this section:**
- Part 1: Enable Reputation Metric Reporting for the Configuration Set (p. 348)
- Part 2: Create an IAM Role (p. 348)
- Part 3: Create the Lambda Function (p. 349)
- Part 4: Re-Enable Email Sending for the Configuration Set (p. 350)
- Part 5: Create an Amazon SNS Topic (p. 350)
- Part 6: Create a CloudWatch Alarm (p. 351)
- Part 7: Test the solution (p. 352)

**Part 1: Enable Reputation Metric Reporting for the Configuration Set**

Before you can configure Amazon SES to automatically pause email sending for a configuration set, you must first enable the export of reputation metrics for the configuration set.

To enable the export of bounce and complaint metrics for the configuration set, complete the steps in the section called "Exporting Reputation Metrics" (p. 219).

**Part 2: Create an IAM Role**

The first step in configuring automatic pausing of email sending is to create an IAM role that can execute the `UpdateConfigurationSetSendingEnabled` API operation.

To create the IAM role

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose Roles.
3. Choose Create role.
4. Under Select type of trusted entity, choose AWS service.
5. Under Choose the service that will use this role, choose Lambda. Choose Next: Permissions.
6. On the Attach permissions policies page, choose the following policies:
   - AWS LambdaBasicExecutionRole
   - AmazonSESFullAccess

   **Tip**
   Use the search box at the top of the list of policies to quickly locate these policies.

   Choose Next: Review.
7. On the Review page, for Name, type a name for the role. Choose Create role.
Part 3: Create the Lambda Function

After you create an IAM role, you can create the Lambda function that pauses email sending for the configuration set.

To create the Lambda function

1. Open the AWS Lambda console at https://console.aws.amazon.com/lambda/.
2. Use the region selector to choose the region in which you want to deploy this Lambda function.
   
   **Note**
   
   This function only pauses email sending for configuration sets in the AWS Region you select in this step. If you send email from more than one region, repeat the procedures in this section for each region in which you want to automatically pause email sending.

3. Choose Create function.
4. Under Create function, choose Author from scratch.
5. Under Author from scratch, complete the following steps:
   
   - For **Name**, type a name for the Lambda function.
   - For **Runtime**, choose Node.js 6.10.
   - For **Role**, choose Choose an existing role.
   - For **Existing role**, choose the IAM role you created in the section called “Part 2: Create an IAM Role” (p. 348).

   Choose Create function.

6. Under **Function code**, in the code editor, paste the following code:

   ```javascript
   'use strict';
   var aws = require('aws-sdk');
   // Create a new SES object.
   var ses = new aws.SES();
   // Specify the parameters for this operation. In this example, you pass the // Enabled parameter, with a value of false (Enabled = false disables email // sending, Enabled = true enables it). You also pass the ConfigurationSetName // parameter, with a value equal to the name of the configuration set for // which you want to pause email sending.
   var params = {
     ConfigurationSetName: 'ConfigSet',
     Enabled: false
   };

   exports.handler = (event, context, callback) => {
     // Pause sending for a configuration set
     ses.updateConfigurationSetSendingEnabled(params, function(err, data) {
       if(err) {
         console.log(err.message);
       } else {
         console.log(data);
       }
     });
   };
   
   Replace ConfigSet in the preceding code with the name of the configuration set. Choose Save.
   ```

349
7. Choose Test. If the Configure test event window appears, type a name in the Event name field, and then choose Create.

8. Ensure that the notification bar at the top of the page says Execution result: succeeded. If the function failed to execute, do the following:

   • Verify that the IAM role you created in the section called “Part 2: Create an IAM Role” (p. 348) contains the correct policies.
   • Verify that the code in the Lambda function does not contain any errors. The Lambda code editor automatically highlights syntax errors and other potential issues.

Part 4: Re-Enable Email Sending for the Configuration Set

A side effect of testing the Lambda function in the section called “Part 3: Create the Lambda Function” (p. 349) is that email sending for the configuration set is paused. In most cases, you do not want to pause sending for the configuration set until the CloudWatch alarm is triggered.

The procedures in this section re-enable email sending for your configuration set. To complete these procedures, you must install and configure the AWS Command Line Interface. For more information, see the AWS Command Line Interface User Guide.

To re-enable email sending

1. At the command line, type the following command to re-enable email sending for the configuration set: `aws ses update-configuration-set-sending-enabled --configuration-set-name ConfigSet --enabled --region us-west-2`

   In the preceding command, replace `ConfigSet` with the name of the configuration set for which you want to pause email sending, and replace `us-west-2` with the region in which you want to automatically pause email sending.

2. At the command line, type the following command to ensure that email sending is enabled: `aws ses describe-configuration-set --configuration-set-name ConfigSet --region us-west-2`

   You will see output similar to the following:

   ```json
   {
     "ConfigurationSet": {
       "Name": "ConfigSet",
     },
     "ReputationOptions": {
       "ReputationMetricsEnabled": true,
       "SendingEnabled": true
     }
   }
   ``

   If the value of SendingEnabled is true, then email sending for the configuration set was successfully re-enabled.

Part 5: Create an Amazon SNS Topic

For CloudWatch to execute the Lambda function when an alarm is triggered, you must first create an Amazon SNS topic and subscribe the Lambda function to it.

To create the Amazon SNS topic

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v2/home.
2. Use the region selector to choose the region in which you want to automatically pause email sending.

3. In the navigation pane, choose Topics.

4. Choose Create new topic.

5. On the Create new topic window, for Topic name, type a name for the topic. Optionally, you can type a more descriptive name in the Display name field.

Choose Create topic.

6. In the list of topics, check the box next to the topic you created in the previous step. On the Actions menu, choose Subscribe to topic.

7. On the Create subscription window, make the following selections:
   
   • For Protocol, choose AWS Lambda.
   
   • For Endpoint, choose the Lambda function you created in the section called “Part 3: Create the Lambda Function” (p. 349).
   
   • For Version or alias, choose default.

Choose Create subscription.

Part 6: Create a CloudWatch Alarm

This section contains procedures for creating an alarm in CloudWatch that is triggered when a metric reaches a certain threshold. When the alarm is triggered, it delivers a notification to the Amazon SNS topic you created in the section called “Part 5: Create an Amazon SNS Topic” (p. 350), which then executes the Lambda function you created in the section called “Part 3: Create the Lambda Function” (p. 349).

To create a CloudWatch alarm


2. Use the region selector to choose the region in which you want to automatically pause email sending.

3. In the navigation pane on the left, choose Alarms.

4. Choose Create Alarm.

5. On the Create Alarm window, under SES Metrics, choose Configuration Set Metrics.

6. In the ses:configuration-set column, locate the configuration set for which you want to create an alarm. Under Metric Name, choose one of the following options:
   
   • Reputation.BounceRate – Choose this metric if you want to pause email sending for the configuration set when the overall hard bounce rate for the configuration set crosses a threshold that you define.
   
   • Reputation.ComplaintRate – Choose this metric if you want to pause email sending for the configuration set when the overall complaint rate for the configuration set crosses a threshold that you define.

Choose Next.

7. Complete the following steps:
   
   • Under Alarm Threshold, for Name, type a name for the alarm.
   
   • Under Whenever: Reputation.BounceRate or Whenever: Reputation.ComplaintRate, specify the threshold that causes the alarm to trigger.
Note
If the overall bounce rate for your Amazon SES account exceeds 10%, or if the overall complaint rate for your Amazon SES account exceeds 0.5%, your Amazon SES account is automatically placed on probation. When you specify the bounce or complaint rate that causes the CloudWatch alarm to trigger, we recommend that you use values that are far below these rates to prevent your account from being placed on probation.

- Under Actions, for Whenever this alarm, choose State is ALARM. For Send notification to, choose the Amazon SNS topic you created in the section called “Part 5: Create an Amazon SNS Topic” (p. 350).

Choose Create Alarm.

Part 7: Test the solution
You can now test the alarm to ensure that it executes the Lambda function when it enters the ALARM state. You can use the SetAlarmState operation in the CloudWatch API to temporarily change the state of the alarm.

The procedures in this section are optional, but we recommend that you complete them to verify that the entire solution is configured correctly.

To test the solution
1. At the command line, type the following command to check the email sending status for the configuration set:
   `aws ses describe-configuration-set --configuration-set-name ConfigSet --region us-west-2`
   If sending is enabled for the configuration set, you see the following output:
   ```
   {
     "ConfigurationSet": {
       "Name": "ConfigSet",
     },
     "ReputationOptions": {
       "ReputationMetricsEnabled": true,
       "SendingEnabled": true
     }
   }
   ```
   If the value of SendingEnabled is true, then email sending is currently enabled for the configuration set.

2. At the command line, type the following command to temporarily change the alarm state to ALARM:
   `aws cloudwatch set-alarm-state --alarm-name MyAlarm --state-value ALARM --state-reason "Testing execution of Lambda function" --region us-west-2`
   Replace MyAlarm in the preceding command with the name of the alarm you created in the section called “Part 6: Create a CloudWatch Alarm” (p. 351).
   
   Note
   When you execute this command, the status of the alarm switches from OK to ALARM and back to OK within a few seconds. You can view these status changes on the alarm’s History tab in the CloudWatch console, or by using the DescribeAlarmHistory operation.

3. At the command line, type the following command to check the email sending status for the configuration set:
   `aws ses describe-configuration-set --configuration-set-name ConfigSet`
   If the Lambda function executed successfully, you see the following output:
If the value of `SendingEnabled` is `false`, then email sending for the configuration set is disabled, indicating that the Lambda function executed successfully.

4. Complete the steps in the section called "Part 4: Re-Enable Email Sending for the Configuration Set" (p. 350) to re-enable email sending for the configuration set.
Controlling Access to Amazon SES

You can use AWS Identity and Access Management (IAM) with Amazon Simple Email Service (Amazon SES) to specify which Amazon SES API actions an IAM user, group, or role can perform. (In this topic we refer to these entities collectively as user.) You can also control which email addresses the user can use for the "From", recipient, and "Return-Path" addresses of emails.

For example, you can create an IAM policy that allows users in your organization to send email, but not perform administrative actions such as checking sending statistics. As another example, you can write a policy that allows a user to send emails through Amazon SES from your account, but only if they use a specific "From" address.

To use IAM, you define an IAM policy, which is a document that explicitly defines permissions, and attach the policy to a user. To learn how to create IAM policies, see the IAM User Guide. Other than applying the restrictions you set in your policy, there are no changes to how users interact with Amazon SES or in how Amazon SES carries out requests.

Note
You can also control access to Amazon SES by using sending authorization policies. Whereas IAM policies constrain what individual IAM users can do, sending authorization policies constrain how individual verified identities can be used. Further, only sending authorization policies can grant cross-account access. For more information about sending authorization, see Using Sending Authorization with Amazon SES (p. 132).

If you are looking for information about how to generate Amazon SES SMTP credentials for an existing IAM user, see Obtaining Your Amazon SES SMTP Credentials (p. 67).

Creating IAM Policies for Access to Amazon SES

This section explains how you can use IAM policies specifically with Amazon SES. To learn how to create IAM policies in general, see the IAM User Guide.

There are three reasons you might use IAM with Amazon SES:

- To restrict the email-sending action.
- To restrict the "From", recipient, and "Return-Path" addresses of the emails that the user sends.
- To control general aspects of API usage such as the time period during which a user is permitted to call the APIs that they are authorized to use.

Restricting the Action

To control which Amazon SES actions a user can perform, you use the Action element of an IAM policy. You can set the Action element to any Amazon SES API action by prefixing the API name with the lowercase string ses:. For example, you can set the Action to ses:SendEmail, ses:GetSendStatistics, or ses:* (for all actions).

Then, depending on the Action, specify the Resource element as follows:

- If the Action element only permits access to email-sending APIs (that is, ses:SendEmail and/or ses:SendRawEmail):
  - To allow the user to send from any identity in your AWS account, set Resource to *
• To limit the identities that the user can send from, set `Resource` to the ARN(s) of the identities that you are permitting the user to use.

**If the `Action` element permits access to all APIs:**

• If you do not want to limit the identities that the user can send from, set `Resource` to *
• If you **do** want to limit the identities that the user can send from, you need to create two policies (or two statements within one policy):
  • One with `Action` set to an explicit list of the permitted non-email-sending APIs and `Resource` set to *
  • One with `Action` set to one of the email-sending APIs (`ses:SendEmail` and/or `ses:SendRawEmail`), and `Resource` set to the ARN(s) of the identities you are permitting the user to use.

For a list of available Amazon SES actions, see the Amazon Simple Email Service API Reference. If the IAM user will be using the SMTP interface, you must allow access to `ses:SendRawEmail` at a minimum.

**Restricting Email Addresses**

If you want to restrict the user to specific email addresses, you can use a `Condition` block. In the `Condition` block, you specify conditions by using condition keys as described in the IAM User Guide. By using condition keys, you can control the following email addresses:

**Note**

These email address condition keys apply only to the APIs noted in the following table.

<table>
<thead>
<tr>
<th>Condition Key</th>
<th>Description</th>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ses:Recipients</code></td>
<td>Restricts the recipient addresses, which include the To:, &quot;CC&quot;, and &quot;BCC&quot; addresses.</td>
<td><code>SendEmail, SendRawEmail</code></td>
</tr>
<tr>
<td><code>ses:FromAddress</code></td>
<td>Restricts the &quot;From&quot; address.</td>
<td><code>SendEmail, SendRawEmail, SendBounce</code></td>
</tr>
<tr>
<td><code>ses:FromDisplayName</code></td>
<td>Restricts the &quot;From&quot; address that is used as the display name.</td>
<td><code>SendEmail, SendRawEmail</code></td>
</tr>
<tr>
<td><code>ses:FeedbackAddress</code></td>
<td>Restricts the &quot;Return-Path&quot; address, which is the address where bounces and complaints can be sent to you by email feedback forwarding. For information about email feedback forwarding, see Amazon SES Notifications Through Email (p. 227).</td>
<td><code>SendEmail, SendRawEmail</code></td>
</tr>
</tbody>
</table>

**Restricting General API Usage**

By using AWS-wide keys in conditions, you can restrict access to Amazon SES based on aspects such as the date and time that user is permitted access to APIs. Amazon SES implements only the following AWS-wide policy keys:
• **aws:CurrentTime**
• **aws:EpochTime**
• **aws:SecureTransport**
• **aws:SourceIp**
• **aws:UserAgent**

For more information about these keys, see the IAM User Guide.

### Example IAM Policies for Amazon SES

This topic provides examples of policies that permit a user access to Amazon SES, but only under certain conditions.

- **Allowing Full Access to All Amazon SES Actions (p. 356)**
- **Allowing Access to Email-Sending Actions Only (p. 356)**
- **Restricting the Time Period of Sending (p. 357)**
- **Restricting the Recipient Addresses (p. 357)**
- **Restricting the "From" Address (p. 357)**
- **Restricting the Display Name of the Email Sender (p. 358)**
- **Restricting the Destination of Bounce and Complaint Feedback (p. 358)**

### Allowing Full Access to All Amazon SES Actions

The following policy allows a user to call any Amazon SES action.

```json
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": ["ses:*"],
        "Resource": "*"
    }]
}
```

### Allowing Access to Email-Sending Actions Only

The following policy permits a user to send email using Amazon SES, but does not permit the user to perform administrative actions such as accessing Amazon SES sending statistics.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ses:SendEmail", "ses:SendRawEmail"],
            "Resource": "*"
        }
    ]
}
```
Restricting the Time Period of Sending

The following policy permits a user to call Amazon SES email-sending APIs only during the month of September 2015.

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Effect": "Allow",
      "Action": ["ses:SendEmail", "ses:SendRawEmail"],
      "Resource": "*",
      "Condition": {
         "DateGreaterThan": {
            "aws:CurrentTime": "2015-08-31T12:00Z"
         },
         "DateLessThan": {
            "aws:CurrentTime": "2015-10-01T12:00Z"
         }
      }
   }
   ]
}
```

Restricting the Recipient Addresses

The following policy permits a user to call the Amazon SES email-sending APIs, but only to recipient addresses in domain example.com.

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Effect": "Allow",
      "Action": ["ses:SendEmail", "ses:SendRawEmail"],
      "Resource": "*",
      "Condition": {
         "ForAllValues:StringLike": {
            "ses:Recipients": ["*@example.com"]
         }
      }
   }
   ]
}
```

Restricting the "From" Address

The following policy permits a user to call the Amazon SES email-sending APIs, but only if the "From" address is marketing@example.com.

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Effect": "Allow",
      "Action": ["ses:SendEmail", "ses:SendRawEmail"],
      "Resource": "*",
      "Condition": {
         "StringEquals": {
```

357
Restricting the Display Name of the Email Sender

The following policy permits a user to call the Amazon SES email-sending APIs, but only if the display name of the “From” address includes Marketing.

```
{  "Version": "2012-10-17",  "Statement": [    {      "Effect": "Allow",      "Action": ["ses:SendEmail", "ses:SendRawEmail"],      "Resource": "+",      "Condition": {        "StringLike": {          "ses:FromDisplayName": "Marketing"        }      }    }  ]}
```

Restricting the Destination of Bounce and Complaint Feedback

The following policy permits a user to call the Amazon SES email-sending APIs, but only if the “Return-Path” of the email is set to feedback@example.com.

```
{  "Version": "2012-10-17",  "Statement": [    {      "Effect": "Allow",      "Action": ["ses:SendBounce"],      "Resource": "+",      "Condition": {        "StringEquals": {          "ses:FromAddress": "bounce@example.com"        }      }    }  ]}
```
"Action": ["ses:SendEmail", "ses:SendRawEmail"],
"Resource": "*",
"Condition": {
  "StringEquals": {
    "ses:FeedbackAddress": "feedback@example.com"
  }
}
Logging Amazon SES API Calls By Using AWS CloudTrail

Amazon SES is integrated with CloudTrail, a service that captures API calls made by or on behalf of Amazon SES in your AWS account and delivers the log files to an Amazon S3 bucket that you specify. CloudTrail captures API calls made from the Amazon SES console or from the Amazon SES API. Using the information collected by CloudTrail, you can determine what request was made to Amazon SES, the source IP address from which the request was made, who made the request, when it was made, and so on. To learn more about CloudTrail, including how to configure and enable it, see the AWS CloudTrail User Guide.

Amazon SES Information in CloudTrail

When CloudTrail logging is enabled in your AWS account, API calls made to a subset of Amazon SES actions are tracked in log files. Amazon SES records are written together with other AWS service records in a log file. CloudTrail determines when to create and write to a new file based on a time period and file size.

The following actions are supported:

- CloneReceiptRuleSet
- CreateReceiptFilter
- CreateReceiptRule
- CreateReceiptRuleSet
- DeleteIdentity
- DeleteIdentityPolicy
- DeleteReceiptFilter
- DeleteReceiptRule
- DeleteReceiptRuleSet
- DeleteVerifiedEmailAddress
- DescribeActiveReceiptRuleSet
- DescribeReceiptRule
- DescribeReceiptRuleSet
- GetIdentityDkimAttributes
- GetIdentityNotificationAttributes
- GetIdentityPolicies
- GetIdentityVerificationAttributes
- GetSendQuota
- GetSendStatistics
- ListIdentities
- ListIdentityPolicies
- ListReceiptFilters
- ListReceiptRuleSets
- ListVerifiedEmailAddresses
PutIdentityPolicy
ReorderReceiptRuleSet
SetActiveReceiptRuleSet
SetReceiptRulePosition
SetIdentityDkimEnabled
SetIdentityFeedbackForwardingEnabled
SetIdentityHeadersInNotificationsEnabled
SetIdentityNotificationTopic
UpdateReceiptRule
VerifyDomainDkim
VerifyDomainIdentity
VerifyEmailAddress
VerifyEmailIdentity

Every log entry contains information about who generated the request. The user identity information in the log helps you determine whether the request was made with root or IAM user credentials, with temporary security credentials for a role or federated user, or by another AWS service. For more information, see the `userIdentity` field in the CloudTrail Event Reference.

You can store your log files in your bucket for as long as you want, but you can also define Amazon S3 lifecycle rules to archive or delete log files automatically. By default, your log files are encrypted by using Amazon S3 server-side encryption (SSE).

You can choose to have CloudTrail publish Amazon SNS notifications when new log files are delivered if you want to take quick action upon log file delivery. For more information, see Configuring Amazon SNS Notifications.

You can also aggregate Amazon SES log files from multiple AWS regions and multiple AWS accounts into a single Amazon S3 bucket. For more information, see Aggregating CloudTrail Log Files to a Single Amazon S3 Bucket.

Understanding Amazon SES Log File Entries

CloudTrail log files contain one or more log entries where each entry is made up of multiple JSON-formatted events. A log entry represents a single request from any source and includes information about the requested action, any parameters, the date and time of the action, and so on. The log entries are not guaranteed to be in any particular order. That is, they are not an ordered stack trace of the public API calls.

The following example shows a CloudTrail log.

```json
{
  "Records": [
    {
      "awsRegion": "us-west-2",
      "eventID": "0ffa308d-1467-4259-8be3-c749753be325",
      "eventName": "DeleteIdentity",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2015-02-02T21:34:50Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.02",
      "recipientAccountId": "111122223333",
      "requestID": "50b87bfe-ab23-11e4-9106-5b36376f9d12",
      "requestParameters": {
```

361
"identity": "amazon.com"
},
"responseElements": null,
"sourceIPAddress": "192.0.2.0",
"userAgent": "aws-sdk-java/unknown-version",
"userIdentity": {
"accessKeyId": "AKIAIOSFODNN7EXAMPLE",
"accountId": "111122223333",
"arn": "arn:aws:iam::111122223333:root",
"principalId": "111122223333",
"type": "Root"
}
},
{ "awsRegion": "us-west-2",
"eventID": "17bb827a-dc8c-4156-90b1-c214e1d35c9",
"eventName": "DeleteVerifiedEmailAddress",
"eventSource": "ses.amazonaws.com",
"eventTime": "2015-02-04T00:57:15Z",
"eventType": "AwsApiCall",
"eventVersion": "1.02",
"recipientAccountId": "111122223333",
"requestID": "c29fb5c1-ac08-11e4-8ff5-a56a3119e253",
"requestParameters": {
"emailAddress": "user@example.com"
},
"responseElements": null,
"sourceIPAddress": "192.0.2.0",
"userAgent": "aws-sdk-java/unknown-version",
"userIdentity": {
"accessKeyId": "AKIAIOSFODNN7EXAMPLE",
"accountId": "111122223333",
"arn": "arn:aws:iam::111122223333:root",
"principalId": "111122223333",
"type": "Root"
}
},
{ "awsRegion": "us-west-2",
"eventID": "0b311e38-b5c6-43b3-9a39-5fbf0c2d0d99",
"eventName": "GetIdentityDkimAttributes",
"eventSource": "ses.amazonaws.com",
"eventTime": "2015-02-02T21:34:50Z",
"eventType": "AwsApiCall",
"eventVersion": "1.02",
"recipientAccountId": "111122223333",
"requestID": "50f92e80-ab23-11e4-9106-5b36376f9d12",
"requestParameters": {
"identities": [
"example.com"
]
},
"responseElements": null,
"sourceIPAddress": "192.0.2.0",
"userAgent": "aws-sdk-java/unknown-version",
"userIdentity": {
"accessKeyId": "AKIAIOSFODNN7EXAMPLE",
"accountId": "111122223333",
"arn": "arn:aws:iam::111122223333:root",
"principalId": "111122223333",
"type": "Root"
}
},
{ "awsRegion": "us-west-2",
"eventID": "bf695be8-1c67-45b0-8f10-fd56afee09dd",
"eventVersion": "1.02",
"recipientAccountId": "111122223333",
"requestID": "50f92e80-ab23-11e4-9106-5b36376f9d12",
"requestParameters": {
"identities": [
"example.com"
]
},
"responseElements": null,
"sourceIPAddress": "192.0.2.0",
"userAgent": "aws-sdk-java/unknown-version",
"userIdentity": {
"accessKeyId": "AKIAIOSFODNN7EXAMPLE",
"accountId": "111122223333",
"arn": "arn:aws:iam::111122223333:root",
"principalId": "111122223333",
"type": "Root"
}
}
"eventName": "GetIdentityNotificationAttributes",
"eventSource": "ses.amazonaws.com",
"eventTime": "2015-02-02T21:34:50Z",
"eventType": "AwsApiCall",
"eventVersion": "1.02",
"recipientAccountId": "111122223333",
"requestID": "5133ed92-ab23-11e4-9106-5b36376f9d12",
"requestParameters": {
    "identities": [
        "example.com"
    ]
},
"responseElements": null,
"sourceIPAddress": "192.0.2.0",
"userAgent": "aws-sdk-java/unknown-version",
"userIdentity": {
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "accountId": "111122223333",
    "arn": "arn:aws:iam::111122223333:root",
    "principalId": "111122223333",
    "type": "Root"
}
},
{
    "awsRegion": "us-west-2",
    "eventID": "8f9aad63-b03a-4d30-a880-33ae0c6b7786",
    "eventName": "GetIdentityVerificationAttributes",
    "eventSource": "ses.amazonaws.com",
    "eventTime": "2015-02-04T00:57:16Z",
    "eventType": "AwsApiCall",
    "eventVersion": "1.02",
    "recipientAccountId": "111122223333",
    "requestID": "c2d23773-ac08-11e4-8ff5-a56a3119e253",
    "requestParameters": {
        "identities": [
            "example.com"
        ]
    },
    "responseElements": null,
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-sdk-java/unknown-version",
    "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "111122223333",
        "arn": "arn:aws:iam::111122223333:root",
        "principalId": "111122223333",
        "type": "Root"
    }
},
{
    "awsRegion": "us-west-2",
    "eventID": "60ef4f01-9826-4fb4-828e-8c36dda81f40",
    "eventName": "GetSendQuota",
    "eventSource": "ses.amazonaws.com",
    "eventTime": "2015-02-04T01:03:27Z",
    "eventType": "AwsApiCall",
    "eventVersion": "1.02",
    "recipientAccountId": "111122223333",
    "requestID": "a0760648-ac09-11e4-8ff5-a56a3119e253",
    "requestParameters": null,
    "responseElements": null,
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-sdk-java/unknown-version",
    "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "111122223333",
}
"arn": "arn:aws:iam::111122223333:root",
"principalId": "111122223333",
"type": "Root"
}
}
{
"awsRegion": "us-west-2",
"eventID": "0fe5eef3-0c28-4480-808e-307b2140a478",
"eventName": "GetSendStatistics",
"eventSource": "ses.amazonaws.com",
"eventTime": "2015-02-02T21:34:51Z",
"eventType": "AwsApiCall",
"eventVersion": "1.02",
"recipientAccountId": "111122223333",
"requestID": "51644c64-ab23-11e4-9106-5b36376f9d12",
"requestParameters": null,
"responseElements": null,
"sourceIPAddress": "192.0.2.0",
"userAgent": "aws-sdk-java/unknown-version",
"userIdentity": {
  "accessKeyId": "AKIAIOSFODN7EXAMPLE",
  "accountId": "111122223333",
  "arn": "arn:aws:iam::111122223333:root",
  "principalId": "111122223333",
  "type": "Root"
}
}
{
"awsRegion": "us-west-2",
"eventID": "6eb8178e-69c3-4a93-8af0-2a5a0f5f209e",
"eventName": "ListIdentities",
"eventSource": "ses.amazonaws.com",
"eventTime": "2015-02-04T01:03:27Z",
"eventType": "AwsApiCall",
"eventVersion": "1.02",
"recipientAccountId": "111122223333",
"requestID": "a0a4de7a-ac09-11e4-8ff5-a56a3119e253",
"requestParameters": {
  "identityType": "Domain",
  "maxItems": 10
},
"responseElements": null,
"sourceIPAddress": "192.0.2.0",
"userAgent": "aws-sdk-java/unknown-version",
"userIdentity": {
  "accessKeyId": "AKIAIOSFODN7EXAMPLE",
  "accountId": "111122223333",
  "arn": "arn:aws:iam::111122223333:root",
  "principalId": "111122223333",
  "type": "Root"
}
}
{
"awsRegion": "us-west-2",
"eventID": "a18a9745-d06a-43e9-aad0-8ee4de50f48",
"eventName": "ListVerifiedEmailAddresses",
"eventSource": "ses.amazonaws.com",
"eventTime": "2015-02-02T21:34:51Z",
"eventType": "AwsApiCall",
"eventVersion": "1.02",
"recipientAccountId": "111122223333",
"requestID": "51ad8a66-ab23-11e4-9106-5b36376f9d12",
"requestParameters": null,
"responseElements": null,
"sourceIPAddress": "192.0.2.0",
"userAgent": "aws-sdk-java/unknown-version",
"userIdentity": {
  "accessKeyId": "AKIAIOSFODN7EXAMPLE",
  "accountId": "111122223333",
  "arn": "arn:aws:iam::111122223333:root",
  "principalId": "111122223333",
  "type": "Root"
}
}
"userIdentity": {
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "accountId": "111122223333",
    "arn": "arn:aws:iam::111122223333:root",
    "principalId": "111122223333",
    "type": "Root"
},

"awsRegion": "us-west-2",
"eventID": "da975f45-e68b-4499-8e3f-31a89140e0c9",
"eventName": "SetIdentityDkimEnabled",
"eventSource": "ses.amazonaws.com",
"eventTime": "2015-02-04T01:01:24Z",
"eventType": "AwsApiCall",
"eventVersion": "1.02",
"recipientAccountId": "111122223333",
"requestID": "5731c4ab-ac09-11e4-bf5f-a56a3119e253",
"requestParameters": {
    "dkimEnabled": true,
    "identity": "example.com"
},
"responseElements": null,
"sourceIPAddress": "192.0.2.0",
"userAgent": "aws-sdk-java/unknown-version",
"userIdentity": {
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "accountId": "111122223333",
    "arn": "arn:aws:iam::111122223333:root",
    "principalId": "111122223333",
    "type": "Root"
},

"awsRegion": "us-west-2",
"eventID": "5d817126-dadb-436f-b480-f9843289f487",
"eventName": "SetIdentityFeedbackForwardingEnabled",
"eventSource": "ses.amazonaws.com",
"eventTime": "2015-02-02T21:34:51Z",
"eventType": "AwsApiCall",
"eventVersion": "1.02",
"recipientAccountId": "111122223333",
"requestID": "51dd4cf8-ab23-11e4-9106-5b36376f9d12",
"requestParameters": {
    "forwardingEnabled": true,
    "identity": "example.com"
},
"responseElements": null,
"sourceIPAddress": "192.0.2.0",
"userAgent": "aws-sdk-java/unknown-version",
"userIdentity": {
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "accountId": "111122223333",
    "arn": "arn:aws:iam::111122223333:root",
    "principalId": "111122223333",
    "type": "Root"
}}
"recipientAccountId": "111122223333",
"requestID": "51dd4cf8-ab23-11e4-9106-5b36376f9d12",
"requestParameters": {
  "enabled": true,
  "identity": "example.com",
  "notificationType": "Bounce"
},
"responseElements": null,
"sourceIPAddress": "192.0.2.0",
"userAgent": "aws-sdk-java/unknown-version",
"userIdentity": {
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
  "accountId": "111122223333",
  "arn": "arn:aws:iam::111122223333:root",
  "principalId": "111122223333",
  "type": "Root"
}
},
{
  "awsRegion": "us-west-2",
  "eventID": "1a31fd43-55ba-4ce7-b3fe-55659e8144c0",
  "eventName": "SetIdentityNotificationTopic",
  "eventSource": "ses.amazonaws.com",
  "eventTime": "2015-02-04T00:59:21Z",
  "eventType": "AwsApiCall",
  "eventVersion": "1.02",
  "recipientAccountId": "111122223333",
  "requestID": "0d553aac-ac09-11e4-8ff5-a56a3119e253",
  "requestParameters": {
    "identity": "example.com",
    "notificationType": "Bounce",
  },
  "responseElements": null,
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "aws-sdk-java/unknown-version",
  "userIdentity": {
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "accountId": "111122223333",
    "arn": "arn:aws:iam::111122223333:root",
    "principalId": "111122223333",
    "type": "Root"
  }
},
{
  "awsRegion": "us-west-2",
  "eventID": "aec73edb-6dac-4503-81bb-cca1102f959e",
  "eventName": "VerifyDomainDkim",
  "eventSource": "ses.amazonaws.com",
  "eventTime": "2015-02-02T21:34:52Z",
  "eventType": "AwsApiCall",
  "eventVersion": "1.02",
  "recipientAccountId": "111122223333",
  "requestID": "52215ada-ab23-11e4-9106-5b36376f9d12",
  "requestParameters": {
    "domain": "example.com"
  },
  "responseElements": {
    "dkimTokens": [
      "3r2ultrqtelopya3v2apjulcvz7z5n5o",
      "yexya47xmy5f3j3e7v966porcmayu6nu",
      "wtilduqduorhmb2vdt2m53yqlc2j2m6tpw"
    ]
  },
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "aws-sdk-java/unknown-version",
"userIdentity": {  
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE",  
  "accountId": "111122223333",  
  "arn": "arn:aws:iam::111122223333:root",  
  "principalId": "111122223333",  
  "type": "Root"  
},  
},  
{  
  "awsRegion": "us-west-2",  
  "eventID": "3b3e2eb7-7ba3-460b-a127-a5f4cedb4469",  
  "eventName": "VerifyDomainIdentity",  
  "eventSource": "ses.amazonaws.com",  
  "eventTime": "2015-02-04T00:59:21Z",  
  "eventType": "AwsApiCall",  
  "eventVersion": "1.02",  
  "recipientAccountId": "111122223333",  
  "requestID": "0d9c2eb7-ac09-11e4-8ff5-a56a3119e253",  
  "requestParameters": {  
    "disableEmailNotifications": false,  
    "domain": "example.com"  
  },  
  "responseElements": {  
    "verificationToken": "pmBGN/7MjnfhTKUZ06Emgq1PeGw81GhcwefcHU="  
  },  
  "sourceIPAddress": "192.0.2.0",  
  "userAgent": "aws-sdk-java/unknown-version",  
  "userIdentity": {  
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",  
    "accountId": "111122223333",  
    "arn": "arn:aws:iam::111122223333:root",  
    "principalId": "111122223333",  
    "type": "Root"  
  }  
},  
{  
  "awsRegion": "us-west-2",  
  "eventID": "eb2e1616-2b7b-4cd2-6d6c-291f83fc1789f",  
  "eventName": "VerifyEmailAddress",  
  "eventSource": "ses.amazonaws.com",  
  "eventTime": "2015-02-02T21:34:53Z",  
  "eventType": "AwsApiCall",  
  "eventVersion": "1.02",  
  "recipientAccountId": "111122223333",  
  "requestID": "5265ddec-ab23-11e4-9106-5b36376f9d12",  
  "requestParameters": {  
    "emailAddress": "user@example.com"  
  },  
  "responseElements": null,  
  "sourceIPAddress": "192.0.2.0",  
  "userAgent": "aws-sdk-java/unknown-version",  
  "userIdentity": {  
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",  
    "accountId": "111122223333",  
    "arn": "arn:aws:iam::111122223333:root",  
    "principalId": "111122223333",  
    "type": "Root"  
  }  
},  
{  
  "awsRegion": "us-west-2",  
  "eventID": "5613b0ff-d6c6-4526-9b53-a603a9231725",  
  "eventName": "VerifyEmailIdentity",  
  "eventSource": "ses.amazonaws.com",  
  "eventTime": "2015-02-04T01:05:33Z",  
  "eventType": "AwsApiCall",  
  "eventVersion": "1.02",  
  "recipientAccountId": "111122223333",  
  "requestID": "5265ddec-ab23-11e4-9106-5b36376f9d12",  
  "requestParameters": {  
    "emailAddress": "user@example.com"  
  },  
  "sourceIPAddress": "192.0.2.0",  
  "userAgent": "aws-sdk-java/unknown-version",  
  "userIdentity": {  
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",  
    "accountId": "111122223333",  
    "arn": "arn:aws:iam::111122223333:root",  
    "principalId": "111122223333",  
    "type": "Root"  
  }  
}
"eventVersion": "1.02",
"recipientAccountId": "111122223333",
"requestID": "eb2ff803-ac09-11e4-8ff5-a56a3119e253",
"requestParameters": {
    "emailAddress": "user@example.com"
},
"responseElements": null,
"sourceIPAddress": "192.0.2.0",
"userAgent": "aws-sdk-java/unknown-version",
"userIdentity": {
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "accountId": "111122223333",
    "arn": "arn:aws:iam::111122223333:root",
    "principalId": "111122223333",
    "type": "Root"
}
Using Credentials With Amazon SES

To interact with Amazon Simple Email Service (Amazon SES), you use security credentials to verify who you are and whether you have permission to interact with Amazon SES. There are different types of credentials, and the credentials you use depend on what you want to do. For example, you use AWS access keys when you send an email using the Amazon SES API, and SMTP credentials when you send an email using the Amazon SES SMTP interface.

The following table lists the types of credentials you might use with Amazon SES, depending on what you are doing.

<table>
<thead>
<tr>
<th>If you want to access the...</th>
<th>Use these credentials</th>
<th>What the credentials consist of</th>
<th>How to get the credentials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon SES API (You might access the Amazon SES API directly, or indirectly through an AWS SDK, the AWS Command Line Interface, or the AWS Tools for Windows PowerShell.)</td>
<td>AWS access keys</td>
<td>Access key ID and secret access key</td>
<td>See Access Keys in the AWS General Reference.</td>
</tr>
<tr>
<td>Amazon SES SMTP interface</td>
<td>SMTP credentials</td>
<td>User name and password</td>
<td>See Obtaining Your Amazon SES SMTP Credentials (p. 67).</td>
</tr>
</tbody>
</table>

**Note**
For security best practice, use AWS Identity and Access Management (IAM) user access keys instead of AWS account access keys. Your AWS account credentials grant full access to all your AWS resources, so you should store them in a safe place and instead use IAM user credentials for day-to-day interaction with AWS. For more information, see Root Account Credentials vs. IAM User Credentials in the AWS General Reference.

Although your Amazon SES SMTP credentials are different than your AWS access keys and IAM user access keys, Amazon SES SMTP credentials are actually a type of IAM credentials. An IAM user can create Amazon
<table>
<thead>
<tr>
<th>If you want to access the...</th>
<th>Use these credentials</th>
<th>What the credentials consist of</th>
<th>How to get the credentials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon SES console</td>
<td>IAM user name and password OR Email address and password</td>
<td>IAM user name and password OR Email address and password</td>
<td>See IAM User Name and Password and Email Address and Password of the AWS General Reference. Note For security best practice, use an IAM user name and password instead of an email address and password. The email address and password combination are for your AWS account, so you should store them in a safe place instead of using them for day-to-day interaction with AWS. For more information, see Root Account Credentials vs. IAM User Credentials in the AWS General Reference.</td>
</tr>
</tbody>
</table>

For more information about different types of AWS security credentials (except for SMTP credentials, which are used only for Amazon SES), see AWS Security Credentials in the AWS General Reference.
Using the Amazon SES API

You can access the Amazon SES API using an AWS SDK, which wraps the low-level functionality of the Amazon SES API with higher-level data types and function calls that take care of the details for you. You can also make raw requests to the Amazon SES Query API over HTTPS. For general information about the Query API, see Amazon SES Query API (p. 371). Individual API operations are described in the Amazon Simple Email Service API Reference.

This section contains the following topics:

- the section called “Query API” (p. 371)
- the section called “API Error Codes” (p. 376)

Amazon SES Query API

This section describes how to make Query requests to Amazon SES. The various topics acquaint you with the Amazon SES Query interface, the components of a request, how to authenticate a request, and the content of responses.

- For information about Query requests, see Query Requests and Amazon SES (p. 371).
- For information about request authentication, see Request Authentication and Amazon SES (p. 373).
- For examples of GET and POST requests, see GET and POST Examples for Amazon SES (p. 373).
- For information about Query responses, see Query Responses and Amazon SES (p. 374).

Query Requests and Amazon SES

Amazon SES supports Query requests for service actions. Query requests are simple HTTPS requests that use the GET or POST method. Query requests must contain an Action parameter to indicate the action to be performed.

Important
For security reasons, Amazon SES does not support HTTP requests. You must use HTTPS instead.

Structure of a GET Request

This guide presents the Amazon SES GET requests as URLs. Each URL consists of the following:

- Endpoint—The resource the request is acting on. For a list of Amazon SES endpoints, see Regions and Amazon SES (p. 409).
- Action—The action you want to perform on the endpoint, such as sending a message.
- Parameters—Any request parameters.

The following is an example GET request to send a message using the Amazon SES endpoint in the US West (Oregon) region.

https://email.us-west-2.amazonaws.com?Action=SendEmail&Source=user%40example.com&Destination.ToAddresses.member.1=allan
Important
Because the GET requests are URLs, you must URL-encode the parameter values. For example, in the preceding example request, the value for the Source parameter is actually user@example.com. However, the "@" character is not allowed in URLs, so each "@" is URL-encoded as "%40".

To make the GET examples easier to read, this guide presents them in the following parsed format.

```
https://email.us-west-2.amazonaws.com
?Action=SendEmail
&Source=user%40example.com
&Destination.ToAddresses.member.1=allan%40example.com
&Message.Subject.Data=This%20is%20the%20subject%20line.
```

The first line represents the endpoint of the request. After the endpoint is a question mark (?), which separates the endpoint from the parameters. Each parameter is separated by an ampersand (&).

The Action parameter indicates the action to perform. For a complete list of actions, and the parameters used with each action, see the Amazon Simple Email Service API Reference.

Some operations take lists of parameters. For example, when you send an email to multiple recipients, you can provide a list of email addresses. You specify this type of list with param.n notation, where values of n are integers starting from 1. For example, you would specify the first "To:" address using Destination.ToAddresses.1, the second with Destination.ToAddresses.2, etc.

In Amazon SES, spaces are not allowed in any of the parameter values. In this guide, any example Query request parameter value that includes spaces is displayed in one of two different ways:

- URL-encoded (as %20).
- Represented by a plus sign (+). Within a Query request, a plus sign is reserved as a shorthand notation for a space. (If you want to include a literal, uninterpreted plus sign in any parameter, you must URL-encode it as %2B.)

Note
Every request must be accompanied by an X-Amzn-Authorization HTTP header. For more information, see Request Authentication and Amazon SES (p. 373).

Structure of a POST Request

Amazon SES also accepts POST requests. With a POST request, you send the query parameters as a form in the HTTP request body as described in the following procedure.

To create a POST request

1. Assemble the query parameter names and values into a form.

Put the parameters and values together as you would for a GET request (with an ampersand separating each name-value pair). The following example shows a SendEmail request with the line breaks we use in this guide to make the information easier to read.

```
Action=SendEmail
&Source=user@example.com
&Destination.ToAddresses.member.1=allan@example.com
```
2. Form-URL-encode the form according to the Form Submission section of the HTML specification.

For more information, see http://www.w3.org/MarkUp/html-spec/html-spec_toc.html#SEC8.2.1.

3. Provide the resulting form as the body of the POST request.

4. Include the following HTTP headers in the request:

   • Content-Type, with the value set to application/x-www-form-urlencoded
   • Content-Length
   • Date
   • X-Amzn-Authorization (For more information, see Request Authentication and Amazon SES (p. 373).)

5. Send the completed request.

   POST / HTTP/1.1
   Date: Thu, 26 May 2011 06:49:50 GMT
   Host: email.us-west-2.amazonaws.com
   Content-Type: application/x-www-form-urlencoded
   X-Amzn-Authorization: AWS3
     AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE,Signature=1BP67vCvGlDMBQ=dofZxg8E8SUEXAMPLE,Algorithm=HmacSHA256
   Content-Length: 230
   Action=SendEmail
   &Source=user%40example.com
   &Destination.ToAddresses.member.1=allan%40example.com
   &Message.Subject.Data=This%20is%20the%20subject%20line.

The X-Amzn-Authorization header you provide is the same header you would provide if you sent a GET request.

Note
Your HTTP client typically adds other items to the HTTP request as required by the version of HTTP that the client uses. We don't include those additional items in the examples in this guide.

Request Authentication and Amazon SES

When you make a request to the Amazon SES API, you must provide proof that you are truly the account holder so that Amazon SES can verify your identity and whether you are registered to use services offered by AWS. If either test fails, Amazon SES returns an error and does not process the request.


GET and POST Examples for Amazon SES

The following are examples of GET and POST requests, using the Query API.
Example GET Request

Here is an example of what a GET request might look like, including the calculated signature. Notice that all of the parameters have been URL-encoded.

```plaintext
https://email.us-west-2.amazonaws.com/
?Action=SendEmail
&Source=user@example.com
&Destination.ToAddresses.member.1=allan@example.com
&Message.Subject.Data=This%20is%20the%20subject%20line.
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SSignature=RhU864jFu89mg7g9N9j9nr6h7EXAMPLE
&Algorithm=HMACSHA256
```

Example POST Request

Here is an example of what a POST request might look like, before calculating the signature. Notice that all of the parameters have been URL-encoded.

```plaintext
POST / HTTP/1.1
Host: email.us-west-2.amazonaws.com
Content-Type: application/x-www-form-urlencoded
Date: Tue, 25 May 2010 21:20:27 +0000
Content-Length: 174
Action=SendRawEmail
&Destinations.member.1=allan@example.com
&RawMessage.Data=RnJvbTp1c2VyQGV4YW1wbGUuY29tDQpTdWJqZWN0OiBUZXN0DQoNCk1lc3 ...
```

The value for `RawMessage.Data` is a base64-encoded representation of the following text.

```
From:user@example.com
Subject: Test
Message sent using SendRawEmail.
```

Following is the complete POST request to `SendRawEmail`, with the `X-Amzn-Authorization` header. None of the headers should be URL-encoded.

```plaintext
POST / HTTP/1.1
Host: email.us-west-2.amazonaws.com
Content-Type: application/x-www-form-urlencoded
Date: Tue, 25 May 2010 21:20:27 +0000
Content-Length: 174
X-Amzn-Authorization: AWS3-HTTPS
  AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE,Algorithm=HMACSHA256,Signature=lBP67vCvGl ...
Action=SendRawEmail
&Destinations.member.1=allan@example.com
&RawMessage.Data=RnJvbTp1c2VyQGV4YW1wbGUuY29tDQpTdWJqZWN0OiBUZXN0DQoNCk1lc3 ...
```

Query Responses and Amazon SES

In response to a Query request, Amazon SES returns an XML data structure that contains the results of the request.
Every Amazon SES response includes a request ID in a RequestId element. The value is a unique string that AWS assigns. If you ever have issues with a particular request, AWS will ask for the request ID to help troubleshoot the issue.

Successful Amazon SES responses also include one or more message IDs. You can think of a message ID as a receipt for an email message that Amazon SES sends. If a message is rejected or bounced, the message ID will appear in any complaint or bounce notifications that you receive; you can then use the message ID to identify any problematic email messages that you have sent, and take corrective action.

Structure of a Successful Response

If the request succeeded, the main response element is named after the action, but with "Response" appended. For example, SendEmailResponse is the response element returned for a successful SendEmail request. This element contains the following child elements:

- ResponseMetadata, which contains the RequestId child element.
- An optional element containing action-specific results. For example, the SendEmailResponse element includes an element called SendEmailResult.

The XML schema describes the XML response message for each Amazon SES action.

The following is an example of a successful response.

```xml
<SendEmailResponse xmlns="https://email.amazonaws.com/doc/2010-03-31/">
  <SendEmailResult>
    <MessageId>000001271b15238a-fd3ae762-2563-11df-8cd4-6d4e828a9ae8-000000</MessageId>
  </SendEmailResult>
  <ResponseMetadata>
    <RequestId>fd3ae762-2563-11df-8cd4-6d4e828a9ae8</RequestId>
  </ResponseMetadata>
</SendEmailResponse>
```
Structure of an Error Response

If a request is unsuccessful, the main response element is called ErrorResponse regardless of the action that was called. This element contains an Error element and a RequestId element. Each Error includes:

- A Type element that identifies whether the error was a receiver or sender error
- A Code element that identifies the type of error that occurred
- A Message element that describes the error condition in a human-readable form
- A Detail element that might give additional details about the error or might be empty

The following is an example of an error response.

```
<ErrorResponse>
  <Error>
    <Type>
      Sender
    </Type>
    <Code>
      Validation Emirates
    </Code>
    <Message>
      Value null at 'message.subject' failed to satisfy constraint: Member must not be null
    </Message>
  </Error>
  <RequestId>
    42d59b56-7407-4c4a-be0f-4c88daeea257
  </RequestId>
</ErrorResponse>
```

API Error Codes Returned by Amazon SES

This topic contains a list of error codes that are returned by the Amazon SES Query (HTTPS) API. For more information about the Amazon SES API, see the Amazon Simple Email Service API Reference.

You should retry HTTPS requests that receive 5xx errors. In this case, to reduce the likelihood of generating duplicates, we recommend that you implement an exponential retry method with progressively longer waits (5, 10, and 30 seconds) between consecutive timeouts. If the third retry call does not succeed, perform another set of retries after 20 minutes. For an example implementation that uses an exponential retry policy with Amazon SES, see How to handle a “Throttling - Maximum sending rate exceeded” error on the Amazon SES blog.

**Note**
AWS SDKs implement retry logic automatically.

HTTPS client errors (4xx) indicate that you need to revise the request to correct the problem before trying again. For example, if your AWS authentication credentials are invalid, you must update your setup to use the proper credentials before trying to send the email again.

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
<th>HTTPS Status Code</th>
<th>Actions That Return This Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConfigurationSetDoesNotExist</td>
<td>The specified configuration set does not exist. A</td>
<td>400</td>
<td>SendEmail, SendRawEmail</td>
</tr>
<tr>
<td>Error</td>
<td>Description</td>
<td>HTTPS Status Code</td>
<td>Actions That Return This Code</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>IncompleteSignature</td>
<td>The request signature does not conform to AWS standards.</td>
<td>400</td>
<td>All</td>
</tr>
<tr>
<td>InternalFailure</td>
<td>The request processing has failed because of an unknown error, exception, or failure.</td>
<td>500</td>
<td>All</td>
</tr>
<tr>
<td>InvalidAction</td>
<td>The requested action or operation is invalid. Verify that the action is typed correctly.</td>
<td>400</td>
<td>All</td>
</tr>
<tr>
<td>InvalidClientTokenId</td>
<td>The X.509 certificate or AWS access key ID provided does not exist in our records.</td>
<td>403</td>
<td>All</td>
</tr>
<tr>
<td>InvalidParameterCombination</td>
<td>Parameters that must not be used together were used together.</td>
<td>400</td>
<td>All</td>
</tr>
<tr>
<td>InvalidParameterValue</td>
<td>An invalid or out-of-range value was supplied for the input parameter.</td>
<td>400</td>
<td>All</td>
</tr>
<tr>
<td>InvalidQueryParameter</td>
<td>The AWS query string is malformed, does not adhere to AWS standards.</td>
<td>400</td>
<td>All</td>
</tr>
<tr>
<td>MailFromDomainNotVerified</td>
<td>The message could not be sent because Amazon SES could not read the MX record required to use the specified MAIL FROM domain.</td>
<td>400</td>
<td>SendEmail, SendRawEmail</td>
</tr>
<tr>
<td>MalformedQueryString</td>
<td>The query string contains a syntax error.</td>
<td>404</td>
<td>All</td>
</tr>
<tr>
<td>Error</td>
<td>Description</td>
<td>HTTPS Status Code</td>
<td>Actions That Return This Code</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>MessageRejected</td>
<td>Indicates that the action failed, and the message could not be sent. Check the error stack for a description of what caused the error. For more information about problems that can cause this error, see Amazon SES Email Sending Errors (p. 423).</td>
<td>400</td>
<td>SendEmail, SendRawEmail</td>
</tr>
<tr>
<td>MissingAction</td>
<td>The request is missing an action or a required parameter.</td>
<td>400</td>
<td>All</td>
</tr>
<tr>
<td>MissingAuthenticationToken</td>
<td>The request must contain either a valid (registered) AWS access key ID or X.509 certificate.</td>
<td>403</td>
<td>All</td>
</tr>
<tr>
<td>MissingParameter</td>
<td>A required parameter for the specified action is not supplied.</td>
<td>400</td>
<td>All</td>
</tr>
<tr>
<td>OptInRequired</td>
<td>The AWS access key ID needs a subscription for the service.</td>
<td>403</td>
<td>All</td>
</tr>
<tr>
<td>RequestExpired</td>
<td>The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.</td>
<td>400</td>
<td>All</td>
</tr>
<tr>
<td>ServiceUnavailable</td>
<td>The request failed due to a temporary failure of the server.</td>
<td>503</td>
<td>All</td>
</tr>
<tr>
<td>Throttling</td>
<td>The request was denied due to request throttling.</td>
<td>400</td>
<td>All</td>
</tr>
</tbody>
</table>
Amazon SES Code Examples

This section contains code examples that help you get started using Amazon SES to send email and manage your Amazon SES account. Code examples are available in the following languages: C#, Go, Java, PHP, Python, and Ruby.

Choose one of the following links to see code examples for that task:
- Sending Email using AWS SDKs (p. 379)
- Sending Email using the Amazon SES SMTP Interface (p. 389)
- Sending Raw Email using AWS SDKs (p. 400)
- Verify Multiple Email Addresses (p. 407)

Sending Email using AWS SDKs

The AWS SDKs contain built-in methods for interacting with Amazon SES and several other AWS services. If you plan to use Amazon SES along with other AWS services, we recommend that you use an SDK. To learn more about the AWS SDKs, see Tools for Amazon Web Services.

In this section, you will find code examples in several programming languages that demonstrate the process of sending email through Amazon SES using the AWS SDKs.

C#

The following code example is a complete solution for sending email through Amazon SES using the AWS SDK for .NET. This code example assumes that you have installed the AWS SDK for .NET, and that you have created a shared credentials file. For more information about creating a shared credentials file, see Create a Shared Credentials File (p. 28).

```csharp
using Amazon;
using System;
using System.Collections.Generic;
using Amazon.SimpleEmail;
using Amazon.SimpleEmail.Model;

namespace AmazonSESSample
{
    class Program
    {
        // Replace sender@example.com with your "From" address.
        // This address must be verified with Amazon SES.
        static readonly string senderAddress = "sender@example.com";

        // Replace recipient@example.com with a "To" address. If your account
        // is still in the sandbox, this address must be verified.
        static readonly string receiverAddress = "recipient@example.com";

        // The configuration set to use for this email. If you do not want to use a
        // configuration set, comment out the following property and the
        // ConfigurationSetName = configSet argument below.
        static readonly string configSet = "ConfigSet";

        // The subject line for the email.
```
static readonly string subject = "Amazon SES test (AWS SDK for .NET)";

// The email body for recipients with non-HTML email clients.
static readonly string textBody = "Amazon SES Test (.NET)\r\n" + "This email was sent through Amazon SES " + "using the AWS SDK for .NET."

// The HTML body of the email.
static readonly string htmlBody = @"<html><head></head><body><h1>Amazon SES Test (AWS SDK for .NET)</h1><p>This email was sent with
<a href='https://aws.amazon.com/ses/'>Amazon SES</a> using the
<a href='https://aws.amazon.com/sdk-for-net/'>AWS SDK for .NET</a></p></body></html>";

static void Main(string[] args)
{
    // Replace USWest2 with the AWS Region you're using for Amazon SES.
    // Acceptable values are EUWest1, USEast1, and USWest2.
    using (var client = new AmazonSimpleEmailServiceClient(RegionEndpoint.USWest2))
    {
        var sendRequest = new SendEmailRequest
        {
            Source = senderAddress,
            Destination = new Destination
            {
                ToAddresses =
                new List<string> { receiverAddress }
            },
            Message = new Message
            {
                Subject = new Content(subject),
                Body = new Body
                {
                    Html = new Content
                    {
                        Charset = "UTF-8",
                        Data = htmlBody
                    },
                    Text = new Content
                    {
                        Charset = "UTF-8",
                        Data = textBody
                    }
                }
            },
            // If you are not using a configuration set, comment
            // or remove the following line
            ConfigurationSetName = configSet
        };
        try
        {
            Console.WriteLine("Sending email using Amazon SES...");
            var response = client.SendEmail(sendRequest);
            Console.WriteLine("The email was sent successfully.");
        }
        catch (Exception ex)
        {
            Console.WriteLine("The email was not sent.");
            Console.WriteLine("Error message: " + ex.Message);
        }
    }
}
Go

The following code example is a complete solution for sending email through Amazon SES using the AWS SDK for Go. This code example assumes that you have installed the AWS SDK for Go, and that you have created a shared credentials file. For more information about creating a shared credentials file, see Create a Shared Credentials File (p. 28).

```go
package main

import {
    "fmt"
    //go get -u github.com/aws/aws-sdk-go
    "github.com/aws/aws-sdk-go/aws"
    "github.com/aws/aws-sdk-go/aws/session"
    "github.com/aws/aws-sdk-go/service/ses"
    "github.com/aws/aws-sdk-go/aws/awserr"
}

const {
    // Replace sender@example.com with your "From" address.
    // This address must be verified with Amazon SES.
    Sender = "sender@example.com"

    // Replace recipient@example.com with a "To" address. If your account
    // is still in the sandbox, this address must be verified.
    Recipient = "recipient@example.com"

    // Specify a configuration set. If you do not want to use a configuration
    // set, comment out the following constant and the
    // ConfigurationSetName: aws.String(ConfigurationSet) argument below
    ConfigurationSet = "ConfigSet"

    // Replace us-west-2 with the AWS Region you're using for Amazon SES.
    AwsRegion = "us-west-2"

    // The subject line for the email.
    Subject = "Amazon SES Test (AWS SDK for Go)"

    // The HTML body for the email.
    HtmlBody = "<h1>Amazon SES Test Email (AWS SDK for Go)</h1><p>This email was sent
    " +
    "<a href='https://aws.amazon.com/ses/'>Amazon SES</a> using the " +
    "<a href='https://aws.amazon.com/sdk-for-go/'>AWS SDK for Go</a>.</p>"

    //The email body for recipients with non-HTML email clients.
    TextBody = "This email was sent with Amazon SES using the AWS SDK for Go."

    // The character encoding for the email.
    CharSet = "UTF-8"
}

func main() {
    // Create a new session and specify an AWS Region.
}
```
```go
sess, err := session.NewSession(&aws.Config{
    Region: aws.String(AwsRegion),
})

// Create an SES client in the session.
svc := ses.New(sess)

// Assemble the email.
input := &ses.SendEmailInput{
    Destination: &ses.Destination{
        CcAddresses: []string{
        },
        ToAddresses: []string{
            aws.String(Recipient),
        },
    },
    Message: &ses.Message{
        Body: &ses.Body{
            Html: &ses.Content{
                Charset: aws.String(CharSet),
                Data: aws.String(HtmlBody),
            },
            Text: &ses.Content{
                Charset: aws.String(CharSet),
                Data: aws.String(TextBody),
            },
        },
        Subject: &ses.Content{
            Charset: aws.String(CharSet),
            Data: aws.String(Subject),
        },
    },
    Source: aws.String(Sender),
    // Comment or remove the following line if you are not using a
    // configuration set
    ConfigurationSetName: aws.String(ConfigurationSet),
}

// Attempt to send the email.
result, err := svc.SendEmail(input)

// Display error messages if they occur.
if err != nil {
    if aerr, ok := err.(awserr.Error); ok {
        switch aerr.Code() {
        case ses.ErrCodeMessageRejected:
        case ses.ErrCodeMailFromDomainNotVerifiedException:
            fmt.Println(ses.ErrCodeMailFromDomainNotVerifiedException,
                aerr.Error())
        case ses.ErrCodeConfigurationSetDoesNotExistException:
            fmt.Println(ses.ErrCodeConfigurationSetDoesNotExistException,
                aerr.Error())
        default:
            fmt.Println(aerr.Error())
        }
    } else {
        // Print the error, cast err to awserr.Error to get the Code and
        // Message from an error.
        fmt.Println(err.Error())
    }
    return
}

fmt.Println("Email Sent!")
fmt.Println(result)
```
Java

The following code example is a complete solution for sending email through Amazon SES using the AWS SDK for Java. This code example assumes that you have installed the AWS SDK for Java, and that you have created a shared credentials file. For more information about creating a shared credentials file, see Create a Shared Credentials File (p. 28).

```java
package com.amazonaws.samples;

import java.io.IOException;
import com.amazonaws.services.simpleemail.*;
import com.amazonaws.services.simpleemail.model.*;
import com.amazonaws.regions.*;

public class AmazonSESSample {

    // Replace sender@example.com with your "From" address.
    // This address must be verified with Amazon SES.
    static final String FROM = "sender@example.com";

    // Replace recipient@example.com with a "To" address. If your account
    // is still in the sandbox, this address must be verified.
    static final String TO = "recipient@example.com";

    // The configuration set to use for this email. If you do not want to use a
    // configuration set, comment the following variable and the
    // .withConfigurationSetName(CONFIGSET); argument below.
    static final String CONFIGSET = "ConfigSet";

    // The subject line for the email.
    static final String SUBJECT = "Amazon SES test (AWS SDK for Java)";

    // The HTML body for the email.
    static final String HTMLBODY = "<h1>Amazon SES test (AWS SDK for Java)</h1>
    + "This email was sent with <a href='https://aws.amazon.com/ses/'>Amazon SES</a> using the <a href='https://aws.amazon.com/sdk-for-java/'>AWS SDK for Java</a>";

    // The email body for recipients with non-HTML email clients.
    static final String TEXTBODY = "This email was sent through Amazon SES using the AWS SDK for Java.";

    public static void main(String[] args) throws IOException {
        try {
            AmazonSimpleEmailService client = AmazonSimpleEmailServiceClientBuilder.standard()
                // Replace US_WEST_2 with the AWS Region you're using for
                // Amazon SES.
                .withRegion(Regions.US_WEST_2).build();
            SendEmailRequest request = new SendEmailRequest()
                .withDestination(new Destination().withToAddresses(TO))
                .withMessage(new Message()
                    .withBody(new Body()
                        .withHtml(new Content()
                            .withCharset("UTF-8").withData(HTMLBODY))
                        .withText(new Content()
                            .withCharset("UTF-8").withData(TEXTBODY))
                    .withSubject(new Content()
                        .withCharset("UTF-8").withData(SUBJECT)))
                .withSource(FROM);
        }
    }
}
```
Sending Email using AWS SDKs

// Comment or remove the next line if you are not using a configuration set.
client.sendEmail(request);
System.out.println("Email sent!");
} catch (Exception ex) {
  System.out.println("The email was not sent. Error message: "+ ex.getMessage());
}

JavaScript

The following code example is a complete solution for sending email through Amazon SES using the AWS SDK for JavaScript in Node.js. This code example assumes that you have installed the SDK for JavaScript in Node.js. You must also create a configuration file that contains your AWS Access Key ID, Secret Access Key, and preferred AWS Region. For more information about creating this file, see Loading Credentials in Node.js from a JSON File.

'use strict';

var aws = require('aws-sdk');

// Provide the full path to your config.json file.
aws.config.loadFromPath('./config.json');

// Replace sender@example.com with your "From" address.
// This address must be verified with Amazon SES.
const sender = "Sender Name <sender@recipient.com>";

// Replace recipient@example.com with a "To" address. If your account
// is still in the sandbox, this address must be verified.
const recipient = "recipient@example.com";

// Specify a configuration set. If you do not want to use a configuration
// set, comment the following variable, and the
// ConfigurationSetName : configuration_set argument below.
const configuration_set = "ConfigSet";

// The subject line for the email.
const subject = "Amazon SES Test (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
const body_text = "Amazon SES Test (SDK for JavaScript in Node.js)\r\n" + "This email was sent with Amazon SES using the " + "AWS SDK for JavaScript in Node.js.";

// The HTML body of the email.
const body_html = `<html>
<head></head>
<body>
<h1>Amazon SES Test (SDK for JavaScript in Node.js)</h1>
<p>This email was sent with
  <a href='https://aws.amazon.com/SES/'>Amazon SES</a> using the
</p>
</body>
</html>`;

// The character encoding for the email.
const charset = "UTF-8";

// Create a new SES object.

var ses = new aws.SES();

// Specify the parameters to pass to the API.
var params = {
    Source: sender,
    Destination: {
        ToAddresses: [recipient ],
    },
    Message: {
        Subject: {
            Data: subject,
            Charset: charset
        },
        Body: {
            Text: {
                Data: body_text,
                Charset: charset
            },
            Html: {
                Data: body_html,
                Charset: charset
            }
        }
    },
    ConfigurationSetName: configuration_set
};

// Try to send the email.
ses.sendEmail(params, function(err, data) {
    // If something goes wrong, print an error message.
    if(err) {
        console.log(err.message);
    } else {
        console.log("Email sent! Message ID: ", data.MessageId);
    }
});

PHP

The following code example is a complete solution for sending email through Amazon SES using the AWS SDK for PHP. This code example assumes that you have installed the AWS SDK for PHP, and that you have created a shared credentials file. For more information about creating a shared credentials file, see Create a Shared Credentials File (p. 28).

```php
<?php

// Replace path_to_sdk_inclusion with the path to the SDK as described in
define('REQUIRED_FILE', 'path_to_sdk_inclusion');

// Replace sender@example.com with your "From" address.
// This address must be verified with Amazon SES.
define('SENDER', 'sender@example.com');

// Replace recipient@example.com with a "To" address. If your account
// is still in the sandbox, this address must be verified.
define('RECIPIENT', 'recipient@example.com');

// Specify a configuration set. If you do not want to use a configuration
// set, comment the following variable, and the
// 'ConfigurationSetName' => CONFIGSET argument below.
define('CONFIGSET', 'ConfigSet');
```
// Replace us-west-2 with the AWS Region you're using for Amazon SES.
define('REGION', 'us-west-2');
define('SUBJECT', 'Amazon SES test (AWS SDK for PHP)');
define('HTMLBODY', '<h1>AWS Amazon Simple Email Service Test Email</h1>.
<p>This email was sent with <a href="https://aws.amazon.com/ses/">Amazon SES</a> using the <a href="https://aws.amazon.com/sdk-for-php/">AWS SDK for PHP</a>.
AWS SDK for PHP\</p>');
define('TEXTBODY', 'This email was send with Amazon SES using the AWS SDK for PHP.');
define('CHARSET', 'UTF-8');
require REQUIRED_FILE;

use Aws\Ses\SesClient;
use Aws\Ses\Exception\SesException;

$client = SesClient::factory(array(
'\version' => 'latest',
'region' => REGION
));

try {
$result = $client->sendEmail([\n'Destination' => [\n'ToAddresses' => [
RECIPIENT,
],\n],
'Message' => [\n'Body' => [\n'Html' => [\n'Charset' => CHARSET,
'Data' => HTMLBODY,
],\n'Text' => [\n'Charset' => CHARSET,
'Data' => TEXTBODY,
],\n],
'Source' => SENDER,
'ConfigurationSetName' => CONFIGSET,
]);
$messageId = $result->get('MessageId');
echo("Email sent! Message ID: $messageId\n");
} catch (SesException $error) {
echo("The email was not sent. Error message: "$error->getAwsErrorMessage()."\n");
}

?>
Python

The following code example is a complete solution for sending email through Amazon SES using the AWS SDK for Python (Boto). This code example assumes that you have installed the AWS SDK for Python (Boto), and that you have created a shared credentials file. For more information about creating a shared credentials file, see Create a Shared Credentials File (p. 28).

```python
import boto3
from botocore.exceptions import ClientError

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
SENDER = "Sender Name <sender@example.com>"

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
RECIPIENT = "recipient@example.com"

# Specify a configuration set. If you do not want to use a configuration
# set, comment the following variable, and the
# ConfigurationSetName=CONFIGURATION_SET argument below.
CONFIGURATION_SET = "ConfigSet"

# If necessary, replace us-west-2 with the AWS Region you're using for Amazon SES.
AWS_REGION = "us-west-2"

# The subject line for the email.
SUBJECT = "Amazon SES Test (SDK for Python)"

# The email body for recipients with non-HTML email clients.
BODY_TEXT = ("This email was sent with Amazon SES using the "
             "AWS SDK for Python (Boto)."
            )

# The HTML body of the email.
BODY_HTML = "Amazon SES Test (SDK for Python)"

# The character encoding for the email.
CHARSET = "UTF-8"

# Create a new SES resource and specify a region.
client = boto3.client('ses', region_name=AWS_REGION)

# Try to send the email.
try:
    # Provide the contents of the email.
    response = client.send_email(
        Destination={
            'ToAddresses': [RECIPIENT],
        },
        Message={
            'Body': {
            },
            'Subject': {
                'Data': SUBJECT
            }
        },
        Source=SENDER,
    )
    if 'ResponseMetadata' in response:
        print(response['ResponseMetadata']['HTTPStatusCode'])
    else:
        print('Error')
```

387
The following code example is a complete solution for sending email through Amazon SES using the AWS SDK for Ruby. This code example assumes that you have installed the AWS SDK for Ruby, and that you have created a shared credentials file. For more information about creating a shared credentials file, see Create a Shared Credentials File (p. 28).

```ruby
require 'aws-sdk'

# Replace sender@example.com with your "From" address. This address must be verified with Amazon SES.
sender = "sender@example.com"

# Replace recipient@example.com with a "To" address. If your account is still in the sandbox, this address must be verified.
recipient = "recipient@example.com"

# Specify a configuration set. If you do not want to use a configuration set, comment the following variable and the configuration_set_name: configsetname argument below.
configsetname = "ConfigSet"

# Replace us-west-2 with the AWS Region you're using for Amazon SES.
awsregion = "us-west-2"

# The subject line for the email.
subject = "Amazon SES test (AWS SDK for Ruby)"

# The HTML body of the email.
htmlbody = "<h1>Amazon SES test (AWS SDK for Ruby)</h1>
<p>This email was sent with <a href="https://aws.amazon.com/ses/">Amazon SES</a> using the <a href="https://aws.amazon.com/sdk-for-ruby/">AWS SDK for Ruby</a>.</p>

# The email body for recipients with non-HTML email clients.
textbody = "This email was sent with Amazon SES using the AWS SDK for Ruby."
```
# Specify the text encoding scheme.
encoding = "UTF-8"

# Create a new SES resource and specify a region
ses = Aws::SES::Client.new(region: awsregion)

# Try to send the email.
begin
  # Provide the contents of the email.
  resp = ses.send_email({
    destination: {
      to_addresses: [recipient, ],
    },
    message: {
      body: {
        html: {
          charset: encoding,
          data: htmlbody,
        },
        text: {
          charset: encoding,
          data: textbody,
        },
      },
      subject: {
        charset: encoding,
        data: subject,
      },
    },
    source: sender,
  },
  )
  puts "Email sent!"
  # Comment or remove the following line if you are not using
  # a configuration set
  configuration_set_name: configsetname,
end

  # If something goes wrong, display an error message.
rescue Aws::SES::Errors::ServiceError => error
  puts "Email not sent. Error message: #{error}"
end

## Sending Email using the Amazon SES SMTP Interface

Several programming languages include standard libraries for sending email using SMTP. You can use these libraries to create email sending applications that are lightweight and highly configurable.

In this section, you will find code examples in several programming languages that demonstrate the process of sending email through Amazon SES using the SMTP interface. Wherever possible, these code examples use standard libraries.

### C#

The following code example is a complete solution for sending email through the Amazon SES SMTP interface using C#. In order to run this code example, you must obtain SMTP credentials; for more information, see Obtaining Your Amazon SES SMTP Credentials (p. 67).
using System;
using System.Net;
using System.Net.Mail;

namespace AmazonSESSample
{
    class Program
    {
        static void Main(string[] args)
        {
            // Replace sender@example.com with your "From" address.
            // This address must be verified with Amazon SES.
            const string FROM = "sender@example.com";
            const string FROMNAME = "Sender Name";

            // Replace recipient@example.com with a "To" address. If your account
            // is still in the sandbox, this address must be verified.
            const string TO = "recipient@example.com";

            // Replace smtp_username with your Amazon SES SMTP user name.
            const string SMTP_USERNAME = "smtp_username";

            // Replace smtp_password with your Amazon SES SMTP user name.
            const string SMTP_PASSWORD = "smtp_password";

            // (Optional) the name of a configuration set to use for this message.
            // If you comment out this line, you also need to remove or comment out
            // the "X-SES-CONFIGURATION-SET" header below.
            const string CONFIGSET = "ConfigSet";

            // If you're using Amazon SES in a region other than US West (Oregon),
            // replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP
            // endpoint in the appropriate Region.
            const string HOST = "email-smtp.us-west-2.amazonaws.com";

            // The port you will connect to on the Amazon SES SMTP endpoint. We
            // are choosing port 587 because we will use STARTTLS to encrypt
            // the connection.
            const int PORT = 587;

            // The subject line of the email
            const string SUBJECT = "Amazon SES test (SMTP interface accessed using C#)";

            // The body of the email
            const string BODY = "<h1>Amazon SES Test</h1>
            <p>This email was sent through the "
            <a href='https://aws.amazon.com/ses'>Amazon SES</a> SMTP interface " +
            "using the .NET System.Net.Mail library.</p>";

            // Create and build a new MailMessage object
            MailMessage message = new MailMessage();
            message.IsBodyHtml = true;
            message.From = new MailAddress(FROM, FROMNAME);
            message.To.Add(new MailAddress(TO));
            message.Subject = SUBJECT;
            message.Body = BODY;
            // Comment or delete the next line if you are not using a configuration set
            message.Headers.Add("X-SES-CONFIGURATION-SET", CONFIGSET);

            // Create and configure a new SmtpClient
            SmtpClient client = new SmtpClient(HOST, PORT);
            // Pass SMTP credentials
client.Credentials = new NetworkCredential(SMTP_USERNAME, SMTP_PASSWORD);
// Enable SSL encryption
client.EnableSsl = true;

// Send the email.
try {
    Console.WriteLine("Attempting to send email..."辏
    client.Send(message);
    Console.WriteLine("Email sent!");
} catch (Exception ex) {
    Console.WriteLine("The email was not sent.");
    Console.WriteLine("Error message: " + ex.Message);
}

// Wait for a key press so that you can see the console output
Console.Write("Press any key to continue...");
Console.ReadKey();

Go

The following code example is a complete solution for sending email through the Amazon SES SMTP interface using Go. In order to run this code example, you must obtain SMTP credentials; for more information, see Obtaining Your Amazon SES SMTP Credentials (p. 67). You must also install the Gomail package.

package main

import (  "fmt"  "gopkg.in/gomail.v2"  // go get gopkg.in/gomail.v2)

const (
    // Replace sender@example.com with your "From" address.  
    // This address must be verified with Amazon SES.  
    Sender = "sender@example.com"
    SenderName = "Sender Name"

    // Replace recipient@example.com with a "To" address. If your account  
    // is still in the sandbox, this address must be verified.  
    Recipient = "recipient@example.com"

    // Replace SmtpUser with your Amazon SES SMTP user name.  
    SmtpUser = "SmtpUser"

    // Replace SmtpPass with your Amazon SES SMTP password.  
    SmtpPass = "SmtpPass"

    // The name of the configuration set to use for this message.  
    // If you comment out or remove this variable, you will also need to  
    // comment out or remove the header below.  
    ConfigSet = "ConfigSet"

    // If you’re using Amazon SES in an AWS Region other than US West (Oregon),  
    // replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP  
    // endpoint in the appropriate region.  
    Host = "email-smtp.us-west-2.amazonaws.com"

    // Replace message body with the text you wish to send.  
    message = "Hello!
    "

    // Replace email address(es) with a recipient address.  
    // Multiple addresses should be separated by a comma.  
    email = "recipient@example.com"

    // Replace subject line with a title for your email message.  
    subject = "email subject"
Port = 587

// The subject line for the email.
Subject = "Amazon SES Test (Gomail)"

// The HTML body for the email.
HtmlBody = "<html><head><title>SES Sample Email</title></head><body>" +
  "<h1>Amazon SES Test Email (Gomail)</h1>" +
  "<p>This email was sent with " +
  "<a href='https://aws.amazon.com/ses/'>Amazon SES</a> using " +
  "the <a href='https://github.com/go-gomail/gomail/'>Gomail " +
  "package</a> for <a href='https://golang.org/'>Go</a></p>" +
  "</body></html>"

// The email body for recipients with non-HTML email clients.
TextBody = "This email was sent with Amazon SES using the Gomail package."

// The tags to apply to this message. Separate multiple key-value pairs
// with commas.
// If you comment out or remove this variable, you will also need to
// comment out or remove the header on line 80.
Tags = "genre=test,genre2=test2"

// The character encoding for the email.
CharSet = "UTF-8"

// Create a new message.
m := gomail.NewMessage()

// Set the main email part to use HTML.
m.SetBody("text/html", HtmlBody)

// Set the alternative part to plain text.
m.AddAlternative("text/plain", TextBody)

// Construct the message headers, including a Configuration Set and a Tag.
m.SetHeaders(map[string][]string{
  "From": {m.FormatAddress(Sender,SenderName)},
  "To": {Recipient},
  "Subject": {Subject},
  // Comment or remove the next line if you are not using a configuration set
  "X-SES-CONFIGURATION-SET": {ConfigSet},
  // Comment or remove the next line if you are not using custom tags
  "X-SES-MESSAGE-TAGS": {Tags},
})

// Send the email.
d := gomail.NewPlainDialer(Host, Port, SmtpUser, SmtpPass)

// Display an error message if something goes wrong; otherwise,
// display a message confirming that the message was sent.
if err := d.DialAndSend(m); err != nil {
  fmt.Println(err)
} else {
  fmt.Println("Email sent!")
}
Java

The following code example is a complete solution for sending email through the Amazon SES SMTP interface using Java. In order to run this code example, you must obtain SMTP credentials; for more information, see Obtaining Your Amazon SES SMTP Credentials (p. 67). You must also download the JavaMail API.

```java
package AmazonSESSample;

import java.util.*;
import javax.mail.*;
import javax.mail.internet.*;

public class AmazonSESSample {
    // Replace sender@example.com with your "From" address.
    // This address must be verified.
    static final String FROM = "sender@example.com";
    static final String FROMNAME = "Sender Name";

    // Replace recipient@example.com with a "To" address. If your account
    // is still in the sandbox, this address must be verified.
    static final String TO = "recipient@example.com";

    // Replace smtp_username with your Amazon SES SMTP user name.
    static final String SMTP_USERNAME = "smtp_username";

    // Replace smtp_password with your Amazon SES SMTP password.
    static final String SMTP_PASSWORD = "smtp_password";

    // The name of the Configuration Set to use for this message.
    // If you comment out or remove this variable, you will also need to
    // comment out or remove the header below.
    static final String CONFIGSET = "ConfigSet";

    // Amazon SES SMTP host name. This example uses the US West (Oregon) Region.
    static final String HOST = "email-smtp.us-west-2.amazonaws.com";

    // The port you will connect to on the Amazon SES SMTP endpoint.
    static final int PORT = 587;

    static final String SUBJECT = "Amazon SES test (SMTP interface accessed using Java)";

    static final String BODY = String.join(
            System.getProperty("line.separator"),
            "<h1>Amazon SES SMTP Email Test</h1>",
            "<p>This email was sent with Amazon SES using the " +
            "<a href='https://github.com/javaee/javamail'>Javamail Package</a> for " +
            "<a href='https://www.java.com'>Java</a>." +
        ");

    public static void main(String[] args) throws Exception {
        // Create a Properties object to contain connection configuration information.
        Properties props = System.getProperties();
        props.put("mail.transport.protocol", "smtp");
        props.put("mail.smtp.port", PORT);
        props.put("mail.smtp.starttls.enable", "true");
        props.put("mail.smtp.auth", "true");

        // Create a Session object to represent a mail session with the specified properties.
        Session session = Session.getDefaultInstance(props);
        
        // ... (rest of the code)
    }
}
```
// Create a message with the specified information.
MimeMessage msg = new MimeMessage(session);
msg.setFrom(new InternetAddress(FROM,FROMNAME));
msg.setRecipient(Message.RecipientType.TO, new InternetAddress(TO));
msg.setSubject(SUBJECT);
msg.setContent(BODY,"text/html");

// Add a configuration set header. Comment or delete the
// next line if you are not using a configuration set
msg.setHeader("X-SES-CONFIGURATION-SET", CONFIGSET);

// Create a transport.
Transport transport = session.getTransport();

// Send the message.
try
{
    System.out.println("Sending...");
    // Connect to Amazon SES using the SMTP username and password you specified
    // above.
    transport.connect(HOST, SMTP_USERNAME, SMTP_PASSWORD);
    // Send the email.
    transport.sendMessage(msg, msg.getAllRecipients());
    System.out.println("Email sent!");
}
catch (Exception ex) {
    System.out.println("The email was not sent.");
    System.out.println("Error message: " + ex.getMessage());
}
finally
{
    // Close and terminate the connection.
    transport.close();
}

Perl

The following code example is a complete solution for sending email through the Amazon SES SMTP interface using Perl. In order to run this code example, you must obtain SMTP credentials; for more information, see Obtaining Your Amazon SES SMTP Credentials (p. 67). You must also install the Email::Sender, Email::MIME, and Try::Tiny modules from CPAN.

#!/usr/bin/perl
use warnings;
use strict;
use Email::Sender::Simple qw(sendmail);
use Email::Sender::Transport::SMTP;
use Email::MIME;
use Try::Tiny;

# Replace sender@example.com with your "From" address.
# This address must be verified.
my $sender = 'Sender name <sender@example.com>';  

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
my $recipient = 'recipient@example.com';

# Replace smtp_username with your Amazon SES SMTP user name.
my $smtp_username = "smtp_username";
# Replace smtp_password with your Amazon SES SMTP password.
my $smtp_password = "smtp_password";

# (Optional) the name of a configuration set to use for this message.
# If you comment out this line, you also need to remove or comment out
# the "X-SES-CONFIGURATION-SET:" header below.
my $configset = "ConfigSet";

# If you're using Amazon SES in an AWS Region other than US West (Oregon),
# replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP
# endpoint in the appropriate region.
my $host = "email-smtp.us-west-2.amazonaws.com";
my $port = 587;

# The subject line of the email.
my $subject = "Amazon SES Test (Perl)";

# The HTML body for the email.
my $htmlbody = <<'END_HTML';
<html>
<head></head>
<body>
<h1>Amazon SES SMTP Email Test</h1>
<p>This email was sent with Amazon SES using the
<a href='https://www.perl.org/'>Perl</a>
<a href='http://search.cpan.org/~rjbs/Email-Sender-1.300031/'>Email::Sender</a> library.</p>
</body>
</html>
END_HTML

# The email body for recipients with non-HTML email clients.
my $textbody = "Amazon SES Test
This message was sent with Amazon SES using the Perl "
"Email::Sender module."

# Create the SMTP transport.
my $transport = Email::Sender::Transport::SMTP->new(
    host => "$host",
    port => "$port",
    ssl => 'starttls',
    sasl_username => "$smtp_username",
    sasl_password => "$smtp_password",
);

# Build a multipart MIME message with an HTML part and a text part.
my $message = Email::MIME->create(
    attributes => {
        content_type => 'multipart/alternative',
        charset => 'UTF-8',
    },
    header_str => [
        From => "$sender",
        To => "$recipient",
        Subject => "$subject",
    ],
    parts => [Email::MIME->create(
        attributes => { content_type => 'text/plain' },
        body => "$textbody",
    ),Email::MIME->create(
        attributes => { content_type => 'text/html' },
        body => "$htmlbody",
    )];
PHP

The following code example is a complete solution for sending email through the Amazon SES SMTP interface using PHP. In order to run this code example, you must obtain SMTP credentials; for more information, see Obtaining Your Amazon SES SMTP Credentials (p. 67). You must also install the PHPMailer package using Composer.

```php
<?php

// Modify the path in the require statement below to refer to the
// location of your Composer autoload.php file.
require 'path_to_sdk_inclusion';

// Instantiate a new PHPMailer
$mail = new PHPMailer;

// Tell PHPMailer to use SMTP
$mail->isSMTP();

// Replace sender@example.com with your "From" address.
// This address must be verified with Amazon SES.
// $mail->setFrom('sender@example.com', 'Sender Name');

// Replace recipient@example.com with a "To" address. If your account
// is still in the sandbox, this address must be verified.
// Also note that you can include several addAddress() lines to send
// email to multiple recipients.
$mail->addAddress('recipient@example.com', 'Recipient Name');

// Replace smtp_username with your Amazon SES SMTP user name.
$mail->Username = 'smtp_username';

// Replace smtp_password with your Amazon SES SMTP password.
$mail->Password = 'smtp_password';

// Specify a configuration set. If you do not want to use a configuration
// set, comment or remove the next line.
$mail->addCustomHeader('X-SES-CONFIGURATION-SET', 'ConfigSet');

// If you're using Amazon SES in a region other than US West (Oregon),
// replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP
// endpoint in the appropriate region.
$mail->Host = 'email-smtp.us-west-2.amazonaws.com';

// The subject line of the email
$mail->Subject = 'Amazon SES test (SMTP interface accessed using PHP)';

// The HTML-formatted body of the email
$mail->Body = '<h1>Email Test</h1>

396
```
This email was sent through the Amazon SES SMTP interface using the PHPMailer class.

```php
// Tells PHPMailer to use SMTP authentication
$mail->SMTPAuth = true;

// Enable TLS encryption over port 587
$mail->SMTPSecure = 'tls';
$mail->Port = 587;

// Tells PHPMailer to send HTML-formatted email
$mail->isHTML(true);

// The alternative email body; this is only displayed when a recipient opens the email in a non-HTML email client. The \r\n represents a line break.
$mail->AltBody = "Email Test\r\nThis email was sent through the Amazon SES SMTP interface using the PHPMailer class."

if(!$mail->send()) {
    echo "Email not sent. " , $mail->ErrorInfo , PHP_EOL;
} else {
    echo "Email sent!" , PHP_EOL;
}
?>
```

**Python**

The following code example is a complete solution for sending email through the Amazon SES SMTP interface using Python. In order to run this code example, you must obtain SMTP credentials; for more information, see Obtaining Your Amazon SES SMTP Credentials (p. 67).

```python
import smtplib
import email.utils
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

# Replace sender@example.com with your "From" address.
# This address must be verified.
SENDER = 'sender@example.com'
SENDERNAME = 'Sender Name'

# Replace recipient@example.com with a "To" address. If your account is still in the sandbox, this address must be verified.
RECIPIENT = 'recipient@example.com'

# Replace smtp_username with your Amazon SES SMTP user name.
USERNAME_SMTP = "smtp_username"

# Replace smtp_password with your Amazon SES SMTP password.
PASSWORD_SMTP = "smtp_password"

# (Optional) the name of a configuration set to use for this message. If you comment out this line, you also need to remove or comment out the "X-SES-CONFIGURATION-SET:" header below.
CONFIGURATION_SET = "ConfigSet"

# If you're using Amazon SES in an AWS Region other than US West (Oregon),
# replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP endpoint in the appropriate region.
HOST = "email-smtp.us-west-2.amazonaws.com"
PORT = 587
```
# The subject line of the email.
SUBJECT = 'Amazon SES Test (Python smtplib)'

# The email body for recipients with non-HTML email clients.
BODY_TEXT = ('This email was sent through the Amazon SES SMTP interface using
the Python smtplib package.
')

# The HTML body of the email.
BODY_HTML = '<html>
<head></head>
<body>
<h1>Amazon SES SMTP Email Test</h1>
<p>This email was sent with Amazon SES using the
<a href='https://www.python.org/'>Python</a>
<a href='https://docs.python.org/3/library/smtplib.html'>smtplib</a> library.</p>
</body>
</html>'

# Create message container - the correct MIME type is multipart/alternative.
msg = MIMEMultipart('alternative')
msg['Subject'] = SUBJECT
msg['From'] = email.utils.formataddr((SENDERNAME, SENDER))
msg['To'] = RECIPIENT
# Comment or delete the next line if you are not using a configuration set
msg.add_header('X-SES-CONFIGURATION-SET',CONFIGURATION_SET)
# Record the MIME types of both parts - text/plain and text/html.
part1 = MIMEText(BODY_TEXT, 'plain')
part2 = MIMEText(BODY_HTML, 'html')
# Attach parts into message container.
# According to RFC 2046, the last part of a multipart message, in this case
# the HTML message, is best and preferred.
msg.attach(part1)
msg.attach(part2)

# Try to send the message.
try:
    server = smtplib.SMTP(HOST, PORT)
    server.ehlo()
    server.starttls()
    server.ehlo()
    server.login(USERNAME_SMTP, PASSWORD_SMTP)
    server.sendmail(SENDER, RECIPIENT, msg.as_string())
    server.close()
    # Display an error message if something goes wrong.
except Exception as e:
    print ('Error: ', e)
else:
    print ('Email sent!

Ruby

The following code example is a complete solution for sending email through the Amazon SES SMTP interface using Ruby. In order to run this code example, you must obtain SMTP credentials; for more information, see Obtaining Your Amazon SES SMTP Credentials (p. 67).

require 'net/smtp'
# Replace sender@example.com with your "From" address.  
# This address must be verified with Amazon SES.  
sender = "sender@example.com"
senderName = "Sender Name"

# Replace recipient@example.com with a "To" address. If your account  
# is still in the sandbox, this address must be verified.  
recipient = "recipient@example.com"

# Replace smtp_username with your Amazon SES SMTP user name.  
smtp_username = "smtp_username"

# Replace smtp_password with your Amazon SES SMTP password.  
smtp_password = "smtp_password"

# (Optional) the name of a configuration set to use for this message.  
# If you comment out this line, you also need to remove or comment out  
# the "X-SES-CONFIGURATION-SET" header below.  
configSet = "ConfigSet"

# If you're using Amazon SES in an AWS Region other than US West (Oregon),  
# replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP  
# endpoint in the appropriate region.  
server = "email-smtp.us-west-2.amazonaws.com"  
port = 587

# The subject line of the email.  
subject = "Amazon SES Test (Ruby Net::SMTP library)"

# Specify the headers and body of the message as a variable.  
message = [
    #Remove the next line if you are not using a configuration set  
    "X-SES-CONFIGURATION-SET: #{configSet}",
    "Content-Type: text/html; charset=UTF-8",
    "Content-Transfer-Encoding: 7bit",
    "From: #{senderName} <#{sender}>",
    "To: #{recipient}",
    "Subject: #{subject}",
    "",
    "<h1>Amazon SES Test (Ruby Net::SMTP library)</h1>",
    "<p>This email was sent with \  
      <a href='https://aws.amazon.com/ses/'>\  
      Amazon SES</a> using the Ruby Net::SMTP library.</p>"
].join("\n")

# Create a new SMTP object called "smtp."  
smtp = Net::SMTP.new(server, port)

# Tell the smtp object to connect using TLS.  
smtp.enable_starttls

# Open an SMTP session and log in to the server using SMTP authentication.  
smtp.start(server, smtp_username, smtp_password, :login)

# Try to send the message.  
begin
    smtp.send_message(message, sender, recipient)
    puts "Email sent!"
rescue => e
    puts e
end
Sending Raw Email using AWS SDKs

The AWS SDKs contain built-in methods for interacting with Amazon SES and several other AWS services. If you plan to use Amazon SES along with other AWS services, we recommend that you use an SDK. To learn more about the AWS SDKs, see Tools for Amazon Web Services.

In this section, you will find code examples in several programming languages that demonstrate the process of sending raw email through Amazon SES using the AWS SDKs.

Java

The following code example shows how to use the JavaMail library and the AWS SDK for Java to compose and send a raw email that contains an HTML part, a text part, and an attachment.

This code example assumes you have installed the AWS SDK for Java, and that you have created a shared credentials file. For more information about creating a shared credentials file, see Create a Shared Credentials File (p. 28).

```java
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.PrintStream;
import java.nio.ByteBuffer;
import java.util.Properties;
// JavaMail libraries. Download the JavaMail API from
// https://javaee.github.io/javamail. After you download the jar file,
// include it in the build path for your project.
import javax.activation.DataHandler;
import javax.activation.DataSource;
import javax.activation.FileDataSource;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.Session;
import javax.mail.internet.AddressException;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeBodyPart;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.MimeMultipart;
import javax.mail.internet.MimeUtility;
// AWS SDK libraries. Download the AWS SDK for Java
// from https://aws.amazon.com/sdk-for-java
import com.amazonaws.regions.Regions;
import com.amazonaws.services.simpleemail.AmazonSimpleEmailService;
import com.amazonaws.services.simpleemail.AmazonSimpleEmailServiceClientBuilder;
import com.amazonaws.services.simpleemail.model.RawMessage;
import com.amazonaws.services.simpleemail.model.SendRawEmailRequest;

public class AmazonSESSample {
    // Replace sender@example.com with your "From" address.
    // This address must be verified with Amazon SES.
    private static String SENDER = "Sender Name <sender@example.com>";

    // Replace recipient@example.com with a "To" address. If your account
    // is still in the sandbox, this address must be verified.
    private static String RECIPIENT = "recipient@example.com";

    // Specify a configuration set. If you do not want to use a configuration
    // set, comment the following variable, and the
    // ConfigurationSetName=CONFIGURATION_SET argument below.
    private static String CONFIGURATION_SET = "ConfigSet";

```
private static String SUBJECT = "Customer service contact info";

private static String ATTACHMENT = "C:\\Users\\sender\\customers-to-contact.xlsx";

private static String BODY_TEXT = "Hello,
" + "Please see the attached file for a list " + "of customers to contact.");

private static String BODY_HTML = "<html>
+ "<head></head>
+ "<body>
+ "<h1>Hello!</h1>
+ "<p>Please see the attached file for a " + "list of customers to contact.</p>
+ "</body>
+ "</html>";

public static void main(String[] args) throws AddressException, MessagingException, IOException {
    String DefaultCharSet = MimeUtility.getDefaultJavaCharset();
    Session session = Session.getDefaultInstance(new Properties());

    MimeMessage message = new MimeMessage(session);
    message.setSubject(SUBJECT, "UTF-8");
    message.setFrom(new InternetAddress(SENDER));
    message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(RECIPIENT));

    MimeMultipart msg_body = new MimeMultipart("alternative");

    MimeBodyPart wrap = new MimeBodyPart();

    MimeBodyPart textPart = new MimeBodyPart();
    textPart.setContent(MimeUtility.encodeText(BODY_TEXT, DefaultCharSet, "B"), "text/plain; charset=UTF-8");
    textPart.setHeader("Content-Transfer-Encoding", "base64");

    MimeBodyPart htmlPart = new MimeBodyPart();
    htmlPart.setContent(MimeUtility.encodeText(BODY_HTML, DefaultCharSet, "B"), "text/html; charset=UTF-8");
    htmlPart.setHeader("Content-Transfer-Encoding", "base64");

    msg_body.addBodyPart(textPart);
    msg_body.addBodyPart(htmlPart);

    // Add the child container to the wrapper object.
wrap.setContent(msg_body);
// Create a multipart/mixed parent container.
MimeMultipart msg = new MimeMultipart("mixed");
// Add the parent container to the message.
message.setContent(msg);

// Add the multipart/alternative part to the message.
msg.addBodyPart(wrap);

// Define the attachment
MimeBodyPart att = new MimeBodyPart();
DataSource fds = new FileDataSource(ATTACHMENT);
att.setDataHandler(new DataHandler(fds));
att.setFileName(fds.getName());
// Add the attachment to the message.
msg.addBodyPart(att);

// Try to send the email.
try {
    System.out.println("Attempting to send an email through Amazon SES "
    +"using the AWS SDK for Java... ");

    // Instantiate an Amazon SES client, which will make the service
    // call with the supplied AWS credentials.
    AmazonSimpleEmailService client =
    AmazonSimpleEmailServiceClientBuilder.standard()
    .withRegion(Regions.US_WEST_2).build();

    // Print the raw email content on the console
    PrintStream out = System.out;
    message.writeTo(out);

    // Send the email.
    ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
    message.writeTo(outputStream);
    RawMessage rawMessage =
    new RawMessage(ByteBuffer.wrap(outputStream.toByteArray()));

    SendRawEmailRequest rawEmailRequest =
    new SendRawEmailRequest(rawMessage)
    .withConfigurationSetName(CONFIGURATION_SET);

    client.sendRawEmail(rawEmailRequest);
    System.out.println("Email sent!");
} catch (Exception ex) {
    System.out.println("Email Failed");
    System.err.println("Error message: " + ex.getMessage());
    ex.printStackTrace();
}
}

The following code example shows how to use the PHPMailer package and the AWS SDK for PHP to
compose and send a raw email that contains an HTML part, a text part, and an attachment.

This code example assumes that you have installed the PHPMailer package using Composer. It
also assumes that you have installed the AWS SDK for PHP, and that you have created a shared
credentials file. For more information about creating a shared credentials file, see Create a Shared Credentials File (p. 28).

```php
<?php
require 'vendor/autoload.php';
use PHPMailer\PHPMailer\PHPMailer;
use Aws\Ses\SesClient;
use Aws\Ses\Exception\SesException;

// Replace sender@example.com with your "From" address.
// This address must be verified with Amazon SES.
$sender = 'sender@example.com';
$sendername = 'Sender Name';

// Replace recipient@example.com with a "To" address. If your account
// is still in the sandbox, this address must be verified.
$recipient = 'recipient@example.com';

// Specify a configuration set.
$configset = 'ConfigSet';

// Replace us-west-2 with the AWS Region you're using for Amazon SES.
$region = 'us-west-2';

$subject = 'List of customers to contact';

$htmlbody = <<<EOD
<html>
<head></head>
<body>
<h1>Hello!</h1>
<p>Please see the attached file for a list of customers to contact.</p>
</body>
</html>
EOD;

$textbody = <<<EOD
Hello,
Please see the attached file for a list of customers to contact.
EOD;

// The full path to the file that will be attached to the email.
$att = 'path/to/customers-to-contact.xlsx';

// Create an SesClient.
$client = SesClient::factory(array(
    'version' => 'latest',
    'region' => $region
));

// Create a new PHPMailer object.
$mail = new PHPMailer;

// Add components to the email.
$mail->setFrom($sender, $sendername);
$mail->addAddress($recipient);
$mail->Subject = $subject;
$mail->Body = $htmlbody;
$mail->AltBody = $textbody;
$mail->addAttachment($att);
$mail->addCustomHeader('X-SES-CONFIGURATION-SET', $configset);

// Attempt to assemble the above components into a MIME message.
if (!$mail->preSend()) {
    echo $mail->ErrorInfo;
}
```
} else {
    // Create a new variable that contains the MIME message.
    $message = $mail->getSentMIMEMessage();
}

// Try to send the message.
try {
    $result = $client->sendRawEmail(
        ['RawMessage' => [
            'Data' => $message
        ]
    );;
    // If the message was sent, show the message ID.
    $messageId = $result->get('MessageId');
    echo("Email sent! Message ID: $messageId"."\n");
} catch (SesException $error) {
    // If the message was not sent, show a message explaining what went wrong.
    echo("The email was not sent. Error message: " .
        $error->getAwsErrorMessage("\n"));
}
?>

Python

The following code example shows how to use the Python email package and the AWS SDK for Python (Boto) to compose and send a raw email that contains an HTML part, a text part, and an attachment.

This code example assumes that you have installed the AWS SDK for Python (Boto), and that you have created a shared credentials file. For more information about creating a shared credentials file, see Create a Shared Credentials File (p. 28).

```python
import os
import boto3
from botocore.exceptions import ClientError
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.application import MIMEApplication

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
SENDER = "Sender Name <sender@example.com>

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
RECIPIENT = "recipient@example.com"

# Specify a configuration set. If you do not want to use a configuration
# set, comment the following variable, and the
# ConfigurationSetName=CONFIGURATION_SET argument below.
CONFIGURATION_SET = "ConfigSet"

# If necessary, replace us-west-2 with the AWS Region you're using for Amazon SES.
AWS_REGION = "us-west-2"

# The subject line for the email.
SUBJECT = "Customer service contact info"

# The full path to the file that will be attached to the email.
ATTACHMENT = "path/to/customers-to-contact.xlsx"

# The email body for recipients with non-HTML email clients.
```
BODY_TEXT = "Hello,
Please see the attached file for a list of customers to contact."

# The HTML body of the email.
BODY_HTML = """"
<html>
<head></head>
<body>
<h1>Hello!</h1>
<p>Please see the attached file for a list of customers to contact.</p>
</body>
</html>
""

# The character encoding for the email.
CHARSET = "utf-8"

# Create a new SES resource and specify a region.
client = boto3.client('ses', region_name=AWS_REGION)

# Create a multipart/mixed parent container.
msg = MIMEMultipart('mixed')
# Add subject, from and to lines.
msg['Subject'] = SUBJECT
msg['From'] = SENDER
msg['To'] = RECIPIENT

# Create a multipart/alternative child container.
msg_body = MIMEMultipart('alternative')

# Encode the text and HTML content and set the character encoding. This step is
# necessary if you're sending a message with characters outside the ASCII range.
textpart = MIMEText(BODY_TEXT.encode(CHARSET), 'plain', CHARSET)
htmlpart = MIMEText(BODY_HTML.encode(CHARSET), 'html', CHARSET)
# Add the text and HTML parts to the child container.
msg_body.attach(textpart)
msg_body.attach(htmlpart)

# Define the attachment part and encode it using MIMEApplication.
att = MIMEApplication(open(ATTACHMENT, 'rb').read())
# Add a header to tell the email client to treat this part as an attachment,
# and to give the attachment a name.
att.add_header('Content-Disposition', 'attachment', filename=os.path.basename(ATTACHMENT))

# Attach the multipart/alternative child container to the multipart/mixed
# parent container.
msg.attach(msg_body)

msg.attach(att)

# Add the attachment to the parent container.
msg.attach(att)

try:
    # Provide the contents of the email.
    response = client.send_raw_email(
        Source=SENDER,
        Destinations=[
            RECIPIENT
        ],
        RawMessage={
            'Data':msg.as_string(),
        },
        ConfigurationSetName=CONFIGURATION_SET
    )
# Display an error if something goes wrong.
except ClientError as e:
    print(e.response['Error']['Message'])
else:
    print("Email sent! Message ID:")
    print(response['ResponseMetadata']['RequestId'])

Ruby

The following code example shows how to use the Ruby MIME gem and the AWS SDK for Ruby to compose and send a raw email that contains an HTML part, a text part, and an attachment.

This code example assumes that you have installed the AWS SDK for Ruby and the MIME gem, and that you have created a shared credentials file. For more information about creating a shared credentials file, see Create a Shared Credentials File (p. 28).

```ruby
require 'base64'  #standard library
require 'aws-sdk' #gem install aws-sdk
require 'mime'    #gem install mime

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
sender = "sender@example.com"
sendername = "Sender Name"

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
recipient = "recipient@example.com"

# Specify a configuration set.
configsetname = "ConfigSet"

# Replace us-west-2 with the AWS Region you're using for Amazon SES.
awsregion = "us-west-2"

# The subject line for the email.
subject = "Customer service contact info"

# The full path to the file that will be attached to the email.
attachment = "path/to/customers-to-contact.xlsx"

# The email body for recipients with non-HTML email clients.
textbody = """Hello,
Please see the attached file for a list of customers to contact.
""

# The HTML body of the email.
htmlbody = """<html>
<head></head>
<body>
<h1>Hello!</h1>
<p>Please see the attached file for a list of customers to contact.</p>
</body>
</html>"

# Create a new MIME text object that contains the base64-encoded content of the
# file that will be attached to the message.
file = MIME::Application.new(Base64::encode64(open(attachment,"rb").read))

# Specify that the file is a base64-encoded attachment to ensure that the
# receiving client handles it correctly.
```
file.transfer_encoding = 'base64'
file.disposition = 'attachment'

# Create a MIME Multipart Mixed object. This object will contain the body of the
# email and the attachment.
msg_mixed = MIME::Multipart::Mixed.new

# Create a MIME Multipart Alternative object. This object will contain both the
# HTML and plain text versions of the email.
msg_body = MIME::Multipart::Alternative.new

# Add the plain text and HTML content to the Multipart Alternative part.
msg_body.add(MIME::Text.new(textbody,'plain'))
msg_body.add(MIME::Text.new(htmlbody,'html'))

# Add the Multipart Alternative part to the Multipart Mixed part.
msg_mixed.add(msg_body)

# Add the attachment to the Multipart Mixed part.
msg_mixed.attach(file, 'filename' => attachment)

# Create a new Mail object that contains the entire Multipart Mixed object.
# This object also contains the message headers.
msg = MIME::Mail.new(msg_mixed)
msg.to = { recipient => nil }
msg.from = { sender => sendername }
msg.subject = subject
msg.headers.set('X-SES-CONFIGURATION-SET',configsetname)

# Create a new SES resource and specify a region
ses = Aws::SES::Client.new(region: awsregion)

# Try to send the email.
begin
  # Provide the contents of the email.
  resp = ses.send_raw_email({
    raw_message: {
      data: msg.to_s
    }
  })
  # If the message was sent, show the message ID.
  puts "Email sent! Message ID: " + resp[0].to_s
  # If the message was not sent, show a message explaining what went wrong.
  rescue Aws::SES::Errors::ServiceError => error
    puts "Email not sent. Error message: #{error}" 
end

Verify Multiple Email Addresses

If you are migrating to Amazon SES from another email-sending solution, you may already have a long
list of email addresses that you want to use to send email. The Python script in this example accepts a
JSON-formatted list of email addresses as an input. The following example shows the structure of the
input file:

```json
[
  {
    "email": "carlos.salazar@example.com"
  },
]```
The following script reads the input file and attempts to validate all of the email addresses contained in the file. This code example assumes that you have installed the AWS SDK for Python (Boto), and that you have created a shared credentials file. For more information about creating a shared credentials file, see `Create a Shared Credentials File (p. 28).`

```python
import json     #Python standard library
import boto3    #sudo pip install boto3
from botocore.exceptions import ClientError

# The full path to the file that contains the identities to be verified.
# The input file must be JSON-formatted. See
# for a sample input file.
FILE_INPUT = '/path/to/identities.json'

# If necessary, replace us-west-2 with the AWS Region you're using for Amazon SES.
AWS_REGION = "us-west-2"

# Create a new SES resource specify a region.
client = boto3.client('ses',region_name=AWS_REGION)

# Read the file that contains the identities to be verified.
with open(FILE_INPUT) as data_file:
    data = json.load(data_file)

# Iterate through the array from the input file. Each time an object named
# 'email' is found, run the verify_email_identity operation against the value
# of that object.
for i in data:
    try:
        response = client.verify_email_identity(
            EmailAddress=i['email']
        )
    except ClientError as e:
        print(e.response['Error']['Message'])
    else:
        print('Verification email sent to ' + i['email'] + '. Request ID: ' +
             response['ResponseMetadata']['RequestId'])
```
Regions and Amazon SES

When you use Amazon Simple Email Service (Amazon SES), you connect to a URL that provides an endpoint for the Amazon SES API or SMTP interface. Amazon SES has endpoints in multiple AWS regions. To reduce network latency, it’s a good idea to choose an endpoint closest to your application.

This topic contains information you need to know when you use Amazon SES endpoints in multiple AWS regions. It discusses the following subjects:

- Amazon SES Endpoints (p. 409)
- Selecting a Region to Use with Amazon SES (p. 410)
- Sandbox and Sending Limit Increases (p. 410)
- Verification (p. 411)
- Easy DKIM Setup (p. 411)
- Suppression List (p. 411)
- Feedback Notifications (p. 411)
- SMTP Credentials (p. 412)
- Sending Authorization (p. 412)
- Custom MAIL FROM Domains (p. 412)
- Email Receiving (p. 412)

For general information about AWS regions, see AWS Regions and Endpoints in the AWS General Reference.

Amazon SES Endpoints

The following sections list the AWS regions in which Amazon SES is available, and the corresponding endpoints for sending and receiving emails.

Email Sending Endpoints

The following table lists the endpoints you use for email sending.

<table>
<thead>
<tr>
<th>Region name</th>
<th>API (HTTPS) endpoint</th>
<th>SMTP endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (N. Virginia)</td>
<td>email.us-east-1.amazonaws.com</td>
<td>email-smtp.us-east-1.amazonaws.com</td>
</tr>
<tr>
<td>US West (Oregon)</td>
<td>email.us-west-2.amazonaws.com</td>
<td>email-smtp.us-west-2.amazonaws.com</td>
</tr>
<tr>
<td>EU (Ireland)</td>
<td>email.eu-west-1.amazonaws.com</td>
<td>email-smtp.eu-west-1.amazonaws.com</td>
</tr>
</tbody>
</table>

Email Receiving Endpoints

The following table lists the endpoints you use for email receiving.
Selecting a Region to Use with Amazon SES

The following sections describe how to select a region depending on which method you use to call Amazon SES.

Amazon SES API

When you use the Amazon SES API, you specify an endpoint in the Query request. That endpoint determines the AWS region you are using. For more information, see Query Requests and Amazon SES (p. 371).

Amazon SES SMTP Interface

The SMTP interface is for email sending only. When you use the SMTP interface, the SMTP endpoint you specify in your code or configuration settings determines the AWS region you are using. For more information, see Connecting to the Amazon SES SMTP Endpoint (p. 70).

Amazon SES Console

When you use the Amazon SES console, you can change the endpoint by clicking the region name in the upper right corner of the navigation bar, as shown in the following screenshot.

Sandbox and Sending Limit Increases

Sandbox status and sending limits apply on a per-region basis. You must request sending limit increases for each region individually. When you open an SES Sending Limits Increase case in Support Center, the form has a menu you use to select the AWS region for which you are requesting a sending limit increase. For more information on increasing your sending limits, see Opening an SES Sending Limits Increase Case (p. 130).
Verification

Before you send email using Amazon SES, you must verify that you own your email address or domain with Amazon SES. Verification status for each region is separate, as described in the following sections.

Email Address Verification

You must verify each sender's email address separately for each region you want to use. For example, if you verify an email address in the US West (Oregon) region, you will be able to send from it when you connect to an Amazon SES endpoint in the US West (Oregon) region, but you will not be able to send from it using an endpoint in the US East (N. Virginia) region until you verify that email address in the US East (N. Virginia) region. For more information about verifying email addresses, see Verifying Email Addresses in Amazon SES (p. 43).

Domain Verification

Like email address verification, domain verification applies to each region separately. You must perform the domain verification procedure for each region in which you want to send from a given domain. For example, if you want to send email from example.com from both the US West (Oregon) region endpoint and the US East (N. Virginia) region endpoint, you must add two TXT records to your DNS settings — one record for each region. You generate these records by using the Amazon SES console with the appropriate region selected, or the Amazon SES API endpoint that corresponds to the region you want. For more information about verifying domains, see Verifying Domains in Amazon SES (p. 52).

Easy DKIM Setup

You must perform the Easy DKIM setup procedure for each region in which you want to use Easy DKIM. That is, for each region, you must use the Amazon SES console or the Amazon SES API to generate TXT records, add the TXT records to your DNS settings, and then use the Amazon SES API or the Amazon SES console to enable DKIM signing for your chosen sending identity (email address or domain) within that region. For more information about setting up Easy DKIM, see Easy DKIM in Amazon SES (p. 117).

Suppression List

Although each region has a separate suppression list, if you remove an address from the suppression list of one region, the address is removed from the suppression list of all regions. You remove addresses from the suppression list by using the Amazon SES console. For more information about the suppression list, see Removing an Email Address from the Amazon SES Suppression List (p. 428).

Feedback Notifications

There are two important points to note about setting up feedback notifications in multiple regions:

- Verified identity settings, such as whether you receive feedback by email or through Amazon Simple Notification Service (Amazon SNS), apply only to the region in which you set them. For example, if you verify user@example.com in the US West (Oregon) and US East (N. Virginia) regions and you want to receive bounced emails via Amazon SNS notifications, you must use the Amazon SES API or the Amazon SES console to set up Amazon SNS feedback notifications for user@example.com in both regions.
- Amazon SNS topics you use for feedback forwarding must be within the same region in which you are using Amazon SES.

SMTP Credentials

You can use the same set of SMTP credentials in all regions. For more information about SMTP credentials, see Obtaining Your Amazon SES SMTP Credentials (p. 67).

Custom MAIL FROM Domains

You can use the same custom MAIL FROM domain for verified identities in different AWS regions. If that is what you want to do, you must still publish only one MX record to the MAIL FROM domain's DNS server. Bounces returned by ISPs will go to the Amazon SES feedback endpoint in the region specified in the MX record first, and then Amazon SES will redirect the bounces to the verified identity in the region that sent the email.

Use the MX record settings that Amazon SES provides during the custom MAIL FROM setup for an identity in one of the regions. The custom MAIL FROM setup process is described in Setting a MAIL FROM Domain (p. 58). For reference, you can find the feedback endpoints for all of the regions in the following table.

<table>
<thead>
<tr>
<th>Region name</th>
<th>Feedback endpoints for custom MAIL FROM sending configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (N. Virginia)</td>
<td>feedback-smtp.us-east-1.amazonaws.com</td>
</tr>
<tr>
<td>US West (Oregon)</td>
<td>feedback-smtp.us-west-2.amazonaws.com</td>
</tr>
<tr>
<td>EU (Ireland)</td>
<td>feedback-smtp.eu-west-1.amazonaws.com</td>
</tr>
</tbody>
</table>

Sending Authorization

The delegate sender must send the emails from the AWS region in which the identity owner's identity is verified. The sending authorization policy that gives permission to the delegate sender must be attached to the identity in that region. For more information about sending authorization, see Using Sending Authorization with Amazon SES (p. 132).

Email Receiving

When you receive email with Amazon SES, all of the resources that you use must be in the same region as the Amazon SES endpoint.

**Note**
For a list of endpoints for Amazon SES email receiving, see Email Receiving Endpoints (p. 409).

For example, if you use the Amazon SES endpoint in US West (Oregon), then any Amazon S3 bucket, Amazon SNS topic, AWS KMS key, and Lambda function that you use must also be in US West (Oregon). Similarly, to receive mail with Amazon SES within a region, you must have an active receipt rule set within that region.
Limits in Amazon SES

This topic lists limits within Amazon Simple Email Service (Amazon SES).

Limits Related to Email Sending

The following tables list limits related to email sending.

Sending Limits

Note
Sending limits are based on recipients rather than on messages.

<table>
<thead>
<tr>
<th>Limit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sending limits in the sandbox environment</td>
<td>• Sending quota: 200 emails per 24-hour period.</td>
</tr>
<tr>
<td></td>
<td>• Maximum send rate: 1 email per second.</td>
</tr>
<tr>
<td></td>
<td>Note</td>
</tr>
<tr>
<td></td>
<td>The rate at which Amazon SES accepts your messages might be less than the maximum send rate.</td>
</tr>
<tr>
<td></td>
<td>To increase your sending limits, open an SES Sending Limit case in Support Center. For more information, see Moving Out of the Amazon SES Sandbox (p. 64).</td>
</tr>
</tbody>
</table>

Message Limits

<table>
<thead>
<tr>
<th>Limit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum message size (including attachments)</td>
<td>10 MB per message (after base64 encoding).</td>
</tr>
<tr>
<td>Accepted header fields</td>
<td>Amazon SES accepts any email headers that follow the format described in RFC 822.</td>
</tr>
<tr>
<td>Accepted attachment types</td>
<td>Amazon SES accepts all file attachment types except for attachments with file extensions listed in Appendix: Unsupported Attachment Types (p. 456).</td>
</tr>
</tbody>
</table>

Sender and Recipient Limits

<table>
<thead>
<tr>
<th>Limit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sender address</td>
<td>Both in and out of the sandbox, you are required to verify the &quot;From&quot;, &quot;Source&quot;, &quot;Sender&quot;, and &quot;Return-Path&quot; email addresses or domains, although not &quot;Reply-To&quot;.</td>
</tr>
</tbody>
</table>
Limits Related to Email Sending Event Publishing

<table>
<thead>
<tr>
<th>Limit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of configuration sets</td>
<td>10,000</td>
</tr>
<tr>
<td>Maximum number of event destinations per configuration set</td>
<td>10</td>
</tr>
<tr>
<td>Maximum number of dimensions per CloudWatch event destination</td>
<td>10</td>
</tr>
</tbody>
</table>

Amazon EC2-Related Limits

<table>
<thead>
<tr>
<th>Limit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email sending over port 25</td>
<td>Amazon EC2 throttles email traffic over port 25 by default. To avoid timeouts when sending email through the Amazon SES SMTP endpoint from Amazon EC2, use a different port (587 or 2587) or fill out a Request to Remove Email Sending Limitations.</td>
</tr>
</tbody>
</table>

Limits Related to Email Receiving

The following table lists limits related to email receiving.

<table>
<thead>
<tr>
<th>Limit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of rules per receipt rule set</td>
<td>100</td>
</tr>
<tr>
<td>Maximum number of actions per receipt rule</td>
<td>10</td>
</tr>
<tr>
<td>Maximum number of recipients per receipt rule</td>
<td>100</td>
</tr>
</tbody>
</table>
General Limits

The following table lists limits that apply to both email sending and email receiving.

### Amazon SES API Limits

<table>
<thead>
<tr>
<th>Limit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate at which you can call Amazon SES API actions</td>
<td>All actions (except for SendEmail and SendRawEmail) are throttled at one request per second. For more information about the Amazon SES API, see the Amazon Simple Email Service API Reference.</td>
</tr>
</tbody>
</table>
Best Practices for Sending Email Using Amazon SES

The way you manage email communications with your customers is referred to as your email program. There are several factors that can lead to the success or failure of your email program; these factors may seem confusing or mysterious at first. However, by understanding how email is delivered, and by following certain best practices, you can increase the chances of your email successfully reaching your customers' inboxes.

This section contains information about the following topics:

- Email Program Success Metrics (p. 416)
- Tips and Best Practices (p. 418)

Email Program Success Metrics

There are several metrics that help measure the success of your email program. This section provides information about the following metrics:

- Bounce Rate (p. 416)
- Complaints (p. 417)
- Message Quality (p. 418)

Bounce Rate

A bounce occurs when an email cannot be delivered to the intended recipient. There are two types of bounces: hard bounces and soft bounces. A hard bounce occurs when the email cannot be delivered because of a persistent issue, such as when an email address does not exist. A soft bounce, occurs when a temporary issue prevents the delivery of an email. Soft bounces can occur when a recipient's inbox is full, or when the receiving server is temporarily unavailable. Amazon Simple Email Service handles soft bounces by attempting to redeliver soft bounced emails for a certain period of time.

It is essential that you monitor the number of hard bounces in your email program, and that you remove hard-bouncing email addresses from your recipient lists. When email receivers detect a high rate of hard bounces, they assume that you don't know your recipients well. As a result, a high hard bounce rate can negatively impact the deliverability of your email messages.

The following guidelines help you avoid bounces and improve your sender reputation:

- Try to keep your hard bounce rate below 5%. The fewer hard bounces in your email program, the more likely ISPs will see your messages as legitimate and valuable. This rate should be considered a reasonable and attainable goal, but is not a universal rule across all ISPs.
- Never rent or buy email lists. These lists may contain large numbers of invalid addresses, which could cause your hard bounce rates to increase dramatically. Furthermore, these lists could contain spam traps—email addresses specifically used to catch illegitimate senders. If your messages land in a spam trap, your delivery rates and sender reputation could be irrevocably damaged.
Complaints

• Keep your list up to date. If you haven't emailed your recipients in a long time, try to validate your customers' statuses through some other means (such as website login activity or purchase history).

• If you don't have a method of verifying your customers' statuses, consider sending a win-back email. A typical win-back email mentions that you haven't heard from the customer in a while, and encourages the customer to confirm that they still want to receive your email. After sending a win-back email, purge all of the recipients who did not respond from your lists.

When you receive bounce, it is vital that you respond to them appropriately by observing the following rules:

• If an email address hard bounces, immediately remove that address from your lists. Do not attempt to re-send messages to hard-bouncing addresses. Repeated hard bounces add up, and ultimately harm your reputation with the recipient's ISP.

• Make sure that your bounce submission address is itself capable of receiving emails.

• If your inbound email comes to you from an ISP, instead of through your own internal servers, an influx of bounce notifications can land in your spam folder or be dropped completely. Ideally, you should not use a hosted email address to receive bounces. If you must, however, then check the spam folder often, and don’t mark the bounce messages as spam. In Amazon SES, you can specify the address to which bounces are sent.

• Usually, a bounce provides the address of the mailbox refusing delivery. However, if you need more granular data to map a recipient address to a particular email campaign, include an X-header with a value you can trace back to your internal tracking system. For more information, see Appendix: Header Fields (p. 455).

Complaints

A complaint occurs when an email recipient clicks the "Mark as Spam" (or equivalent) button in their web-based email client. If you accumulate a large number of these complaints, the ISP assumes that you are sending spam. This has a negative impact on your deliverability rate and sender reputation. Some, but not all, ISPs will notify you when a complaint is reported; this is known as a feedback loop. Amazon SES automatically forwards complaints from ISPs that offer feedback loops to you.

The following guidelines help you avoid complaints and improve your sender reputation:

• Try to keep your complaint rate below 0.1%. The fewer complaints in your email program, the more likely ISPs will see your messages as legitimate and valuable. This rate should be considered a reasonable and attainable goal, but is not a universal rule across all ISPs.

• If a customer complains about a marketing email, you should immediately stop sending that customer marketing emails. However, if your email program also includes other types of emails (such as notification or transactional emails), it may be acceptable to continue to send those types of messages to the recipient who issued the complaint.

• As with hard bounces, if you have a list that you haven't sent email to in a while, ensure that your recipients understand why they're receiving your messages. We recommend that you send a welcome message reminding them of who you are and why you're contacting them.

When you receive complaints, it is vital that you respond to them appropriately by observing the following rules:

• As with bounce, make sure that your complaint submission address is itself capable of receiving emails.

• Make sure that your complaint notifications are not being marked as spam by your ISP or mail system.

• Complaint notifications usually contain the email content; this is different from bounce notifications, which typically only include the email headers. However, in complaint notifications, the email address
of the recipient who complained is typically removed. Use custom X-headers or special identifiers embedded in the email body so that you can identify the email address that filed the complaint. This technique enables you to remove the address from your recipient lists.

**Message Quality**

Email receivers use *content filters* to detect certain attributes in your messages to identify whether your message is legitimate. These content filters automatically review the content of your messages to identify common traits of unwanted to malicious messages. Amazon Simple Email Service uses content filtering technologies to help detect and block messages that contain malware before they are sent.

If an email receiver's content filters determine that your message contains the characteristics of spam or malicious email, your message will most likely be flagged and diverted from recipients' inboxes.

Remember the following when designing your email:

- Modern content filters are intelligent, continuously adapting and changing. They don't rely on a predefined set of rules. Third-party services such as ReturnPath or Litmus can help identify content in your email that could trigger content filters.
- If your email contains links, check the URLs for those links against blacklists, such as those found at URIBL.com and SURBL.org.
- Avoid using link shorteners. Malicious senders may use link shorteners to hide the actual destination of the link. When ISPs notice that link shortening services—even the most reputable ones—are being used for nefarious purposes, they may blacklist those services altogether. If your email contains a link to a blacklisted link shortening service, it won't reach your customers' inboxes, and the success of your email campaign suffers.
- Test every link in your email to ensure that it points to the intended page.
- Make sure your website includes Privacy Policy and Terms of Use documents, and that these documents are up to date. It is a good practice to link to these documents from each email you send. Providing links to these documents demonstrates that you have nothing to hide from your customers, which can foster a relationship of trust.
- If you plan to send high-frequency content (such as "daily deals" messages), ensure that the content of your email is different with each deployment. When you send messages with high frequency, you must ensure that those messages are timely and relevant, rather than repetitive and annoying.

**Tips and Best Practices**

Even when you have your customers' best interests in mind, you may still encounter situations that impact the deliverability of your messages. The following sections contain recommendations to help ensure that your email communications reach your intended audience.

**General Recommendations**

- Put yourself in your customer's shoes. Ask yourself if the message you are sending is something you would want to receive in your own inbox. If the answer is anything less than an enthusiastic "yes!" then you probably shouldn't send it.
- Some industries have a reputation for poor quality or even malicious email practices. If you are involved in the following industries, you must monitor your reputation very closely and resolve issues immediately:
  - Home mortgage
  - Credit
  - Pharmaceuticals and supplements
Domain and "From" Address Considerations

- Think carefully about the addresses you send email from. The "From" address is one of the first pieces of information your recipients see, and therefore can leave a lasting first impression. Additionally, some ISPs associate your reputation with your "From" address.

- Consider using subdomains for different types of communications. For example, assume you are sending email from the domain example.com, and you plan to send both marketing and transactional messages. Rather than sending all of your messages from example.com, send your marketing messages from a subdomain such as marketing.example.com, and your transactional messages from a subdomain such as orders.example.com. Unique subdomains develop their own reputations. Using subdomains reduces the risk of damage to your reputation if, for example, your marketing communications land in a spam trap or trigger a content filter.

- If you plan to send a large number of messages, don't send those messages from an ISP-based address such as sender@hotmail.com. If an ISP notices a large volume of messages coming from sender@hotmail.com, that email is treated differently than an email that comes from an outbound email sending domain that you own.

- Work with your domain registrar to ensure that the WHOIS information for your domain is accurate. Maintaining an honest and up-to-date WHOIS record demonstrates that you value transparency, and allows users to quickly identify whether or not your domain is legitimate.

- Avoid using a no-reply address, such as no-reply@example.com, as your "From" or "Reply-to" address. Using a no-reply@ email address sends your recipients a clear message: that you aren't offering them a way to contact you, and that you're not interested in their feedback.

Authentication

- Authenticate your domain with Authenticating Email with SPF in Amazon SES (p. 116) and SenderID. These authentication methods confirm to email recipients that each email you send is actually from the domain it claims to be from.

- Sign your outbound mail with Authenticating Email with DKIM in Amazon SES (p. 117). This step confirms to recipients that the content has not been changed in transit between sender and receiver.

- You can test your authentication settings for both SPF and DKIM by sending an email to an ISP-based email address that you own, such as a personal Gmail or Hotmail account, and then viewing the message's headers. The headers indicate whether your attempts to authenticate and sign the message were successful.

Building and Maintaining Your Lists

- Implement a double opt-in strategy. When users sign up to receive email from you, send them a message with a confirmation link, and do not start sending them email until they confirm their address by clicking that link. A double opt-in strategy helps reduce the number of hard bounces resulting from typographical errors.

- When collecting email addresses with a web-based form, perform minimal validation on those addresses upon submission. For example, ensure that the addresses you collect are well-formed (that is, they are in the format recipient@example.com), and that they refer to domains with valid MX records.
Compliance

- Use caution when allowing user-defined input to be passed to Amazon SES unchecked. Forums registrations and form submissions present unique risks because the content is completely user-generated, and spammers can fill out forms with their own content. It is your responsibility to ensure that you only send email with high-quality content.

- It is highly unlikely that a standard alias (such as postmaster@, abuse@, or noc@) will ever sign up for your email intentionally. Ensure that you are only sending messages to real people who actually want to receive them. This rule is especially true for standard aliases, which are customarily reserved for email watchdogs. These aliases can be maliciously added to your list as a form of sabotage, in order to damage your reputation.

Compliance

- Be aware of the email marketing and anti-spam laws and regulations in the localities to which you are sending email. You alone are responsible for abiding by these laws. This guide does not cover these laws, so it is important that you take the time to research all applicable laws.

- Laws and regulations related to email marketing around the world include, but are not limited to, the following:
  - United States: CAN-SPAM Act of 2003
  - European Union:
    - Privacy and Electronic Communications Directive (Directive 2002/58/EC)
    - General Data Protection Regulation (Regulation 2016/679) (takes effect May 25, 2018)
  - United Kingdom: PECR
  - Canada: CASL
  - Japan: Act on Regulation of Transmission of Specific Electronic Mail
  - Australia: Spam Act 2003
  - South Korea: Act on Promotion of Information and Communication and Communications Network Utilization and Information Protection of 2001

For more information, see Email Spam Legislation by Country on Wikipedia.
Troubleshooting Amazon SES

This section contains the following topics that may help you when you encounter problems:

- For a list of common delivery problems that you might encounter when you send email, along with corrective actions that you can take, see Amazon SES Delivery Problems (p. 422).
- For a description of issues recipients may see when they receive an email that was sent through Amazon SES, see Problems with Emails Received from Amazon SES (p. 422).
- For a list of errors that can occur when you send an email with Amazon SES, see Amazon SES Email Sending Errors (p. 423).
- For information about domain verification problems that you might encounter, see Amazon SES Email Address and Domain Verification Problems (p. 424).
- For solutions to Easy DKIM issues, see Amazon SES DKIM Problems (p. 427).
- For solutions to problems with bounce, complaint, and delivery notifications, see Amazon SES Notification Problems (p. 428).
- For information about how to remove an email address from the suppression list, see Removing an Email Address from the Amazon SES Suppression List (p. 428).
- For tips on how to increase your email sending speed when you make multiple calls to Amazon SES using either the API or the SMTP interface, see Increasing Throughput with Amazon SES (p. 429).
- For solutions to common problems that you might encounter when you use Amazon SES through its Simple Mail Transfer Protocol (SMTP) interface, see Amazon SES SMTP Issues (p. 430).
- For a list of SMTP response codes that a client application can receive from Amazon SES, see SMTP Response Codes Returned by Amazon SES (p. 433).
- For a list of error codes that are returned by the Amazon SES Query (HTTPS) API, see API Error Codes Returned by Amazon SES (p. 376).
- For a description of common enforcement issues and how to handle them, see Amazon SES Enforcement FAQs (p. 436).
- For a discussion about how IP blacklists affect your sending with Amazon SES, see Amazon SES IP Blacklist FAQs (p. 449).

If you are calling the Amazon SES API directly, see the Amazon Simple Email Service API Reference for the HTTP errors that you might receive.

General Amazon SES Issues

The information on this page will explain and help diagnose issues that you may encounter when using Amazon SES.

Changes that I make are not immediately visible

As a service that is accessed through computers in data centers around the world, Amazon SES uses a distributed computing model called eventual consistency. Any change that you make in Amazon SES (or other AWS services) takes time to become visible from all possible endpoints. Some of the delay results from the time it takes to send the data from server to server and from region to region around the world. In the majority of cases, this delay will be no more than a few minutes.

Some areas in which you may notice a delay include:
Creating and modifying configuration sets – When you create or modify a configuration set (for example, if you associate a dedicated IP pool with an existing configuration set (p. 218)), there may be a brief delay from the time that you create or modify it to the time those changes are active.

Creating and modifying event destinations – When you create or modify an event destination (for example, to tell Amazon SES to send your email sending data to another AWS service (p. 247)), there may be a delay between the time your created or modified the event destination and the time email sending events actually arrive at the specified destination.

Amazon SES Delivery Problems

After you make a successful request to Amazon SES, your message is often sent immediately. At other times, there might be a short delay. In any case, you can be assured that your email will be sent.

When Amazon SES sends your message, however, several factors can prevent it from being delivered successfully, and in some cases you will become aware that delivery failed only when the message you send does not arrive. Use the following process to resolve this situation.

If an email does not arrive, try the following:

- Verify that you made a SendEmail or SendRawEmail request for the email in question and that you received a successful response. (See Structure of a Successful Response (p. 375) for an example.) If you are making these requests programmatically, check your software logs to ensure that the program made the request and received a successful response.
- Read the blog article Three places where your email could get delayed when sending through SES because the problem might actually be a delay rather than a nondelivery.
- Check the sender's email address (the "From" address) to verify that it is valid. Also check the Return-Path address, which is where bounce messages are sent. If your mail bounced, there will be an explanatory error message there.
- Check the AWS Service Health Dashboard to confirm that there is not a known problem with Amazon SES.
- Contact the email recipient or the recipient's ISP. Verify that the recipient is using the correct email address, and inquire whether there have been any known delivery problems with the recipient's ISP. Also, determine whether the email did arrive but was filtered as spam.
- If you have signed up for a paid AWS Support Plan, you can open a new technical support case. In your correspondence with us, please provide any relevant recipient addresses, along with any request IDs or message IDs returned from the SendEmail or SendRawEmail responses.
- Wait to see if the problem is actually a delay, not a permanent delivery failure. To combat spammers, some ISPs temporarily reject incoming messages from unknown sending mail servers. This process, called greylisting, can cause a delay in delivery. Amazon SES will retry these messages. If greylisting is the issue, the ISP might accept the email on one of these retry attempts.

Problems with Emails Received from Amazon SES

The following issue can arise when a recipient receives an email sent through Amazon SES. If you are looking for troubleshooting information that talks about when a recipient does not receive an email at all, see Amazon SES Delivery Problems (p. 422).

- A recipient's email client displays "sent via amazonses.com" as the source of the email—Some email clients display the "via" domain when the sender's domain does not match the domain that the email was actually sent from (in this case, amazonses.com). For more information on why, see this explanation from Google. As a workaround, you can set up Domain Keys Identified Mail (DKIM), which is good practice anyway. When you authenticate your emails using DKIM, email clients will typically not show the "via" domain because the DKIM signature shows that the email is from the domain it claims
to be from. For information about how to set up DKIM, see Authenticating Email with DKIM in Amazon SES (p. 117).

- **Your email is not displaying correctly in a recipient's email client**
  - If your email contains non-ASCII characters, you must construct the email in Multipurpose Internet Mail Extensions (MIME) format and send it using the SendRawEmail API. For more information, see Sending Raw Email Using the Amazon SES API (p. 100).
  - Your email might contain improperly formatted MIME. Ensure that it complies with RFC 2047. For example, it must use appropriate header fields and message body encoding.
  - The recipient's email server or email client might impose limitations on the rendered content.

## Amazon SES Email Sending Errors

This topic reviews the types of email sending-specific errors that you may encounter when you send an email through Amazon SES. If you try to send an email through Amazon SES and the call to Amazon SES fails, Amazon SES returns an error message to your application and does not send the email. The way that you observe this error message depends on the way that you call Amazon SES.

- If you call the Amazon SES API directly, the Query action will return an error. The error may be MessageRejected or one of the errors specified in the Common Errors topic of the Amazon Simple Email Service API Reference.

- If you call Amazon SES using an AWS SDK that uses a programming language that supports exceptions, Amazon SES may throw an exception. The type of exception depends on the SDK and on the error. For example, the exception could be an Amazon SES MessageRejectedException (the actual name may vary depending on the SDK) or a general AWS exception. Regardless of the type of exception, the error type and the error message in the exception will give you more information.

- If you call Amazon SES through its SMTP interface, the way that you experience the error depends on the application. Some applications may display a specific error message, some may not. For a list of SMTP response codes, see SMTP Response Codes Returned by Amazon SES (p. 433).

  **Note**
  When your call to Amazon SES to send an email fails, you are not billed for that email.

The following are the types of Amazon SES-specific problems that can cause Amazon SES to return an error when you try to send an email. These errors are in addition to general AWS errors like MalformedQueryString as specified in the Common Errors topic of the Amazon Simple Email Service API Reference.

- **Email address is not verified. The following identities failed the check in region `<region>`: `<identity1>`, `<identity2>`, `<identity3>`**—You are trying to send email from an email address or domain that you have not verified with Amazon SES (p. 43). This error could apply to the "From", "Source", "Sender", or "Return-Path" address. If your account is still in the sandbox, you also must verify every recipient email address except for the recipients provided by the Amazon SES mailbox simulator (p. 162). If Amazon SES is not able to show all of the failed identities, the error message ends with an ellipsis.

  **Note**
  Amazon SES has endpoints in multiple AWS regions (p. 409), and email address verification status is separate for each AWS region. You must complete the verification process for each sender in the AWS region(s) you want to use.

- **Customer is suspended**—Your AWS account has been blocked from sending email using Amazon SES. You can still access the Amazon SES console and perform any activity (e.g., view your metrics) except for email sending; if you attempt to send an email, you will receive this error message.
If this happens, you should have received an email from Amazon SES to the email address associated with your AWS account informing you of the problem. To appeal your suspension and reinstate email sending privileges, follow the instructions in the email. You will need to explain in detail why you believe that the suspension itself was an error, or the changes you have made to ensure that the same problem does not occur again.

- **Throttling**—Amazon SES is limiting the rate at which you can send messages. Your application may be trying to send too much email, or to send email at too fast a rate. In these cases, the error may be similar to the following:
  - **Daily message quota exceeded**—You have sent the maximum number of messages that you are permitted in a 24-hour period. If you have exceeded your daily quota, you will have to wait until the next 24-hour period before you can send more email.
  - **Maximum sending rate exceeded**—You are attempting to send more emails per second than is permitted by your maximum send rate. If you have exceeded your sending rate, you can continue to send email, but will need to reduce your send rate. For more information, see How to handle a “Throttling - Maximum sending rate exceeded” error on the Amazon SES blog.

You should regularly monitor your sending activity to see how close you are to your sending limits. For more information, see Monitoring Your Amazon SES Sending Limits (p. 128). For general information about sending limits, see Managing Your Amazon SES Sending Limits (p. 127). For information about how to increase your sending limits, see Increasing Your Amazon SES Sending Limits (p. 129).

**Important**

If the error text that explains the throttling error is not related to you exceeding your daily quota or maximum send rate, then there might be a system-wide problem that is causing reduced sending capabilities. For information about the service status, go to the AWS Service Health Dashboard.

- **There are no recipients specified**—No recipients were provided.
- **There are non-ASCII characters in the email address**—The email address string must be 7-bit ASCII. If you want to send to or from email addresses that contain Unicode characters in the domain part of an address, you must encode the domain using Punycode. Punycode is not permitted in the local part of the email address (i.e., the part before the @) nor in the "friendly from" name. If you want to use Unicode characters in the "friendly from" name, you must encode the “friendly from” name using MIME encoded-word syntax, as described in Sending Raw Email Using the Amazon SES API (p. 100). For more information about Punycode, see RFC 3492.
- **Mail FROM domain is not verified**—Amazon SES could not read the MX record required to use the specified MAIL FROM domain. For information about editing the custom MAIL FROM domain settings for an identity, see Editing a MAIL FROM Domain with Amazon SES (p. 61).
- **Configuration set does not exist**—The configuration set that you specified does not exist. A configuration set is an optional parameter that you use to publish email sending events. For more information, see Monitoring Using Amazon SES Event Publishing (p. 247).

## Amazon SES Email Address and Domain Verification Problems

To verify an email address or domain with Amazon SES, you initiate the process using either the Amazon SES console or the Amazon SES API. This section contains information that may help resolve issues with the verification process.
Common Email Verification Problems

The Verification Email Did Not Arrive

If you complete the procedures in Verifying Email Addresses in Amazon SES (p. 43) but do not receive the verification email within a few minutes, complete the following troubleshooting tasks:

- Check the spam or junk mail folder in your email client.
- Confirm that the address you are trying to verify is able to receive email. Send a test email to the address that you want to verify from a separate email address.
- Check the list of verified addresses in the Amazon SES console. Confirm that there are no errors in the email address you want to verify.

Common Domain Verification Problems

If you attempt to verify a domain using the procedure in Verifying Domains in Amazon SES (p. 52) and you encounter problems, review the possible causes and solutions below.

- Your DNS provider does not allow underscores in TXT record names—You can omit _amazonses from the TXT record name.
- You want to verify the same domain multiple times and you can't have multiple TXT records with the same name—You might need to verify your domain more than once because you're sending in different regions or you're sending from multiple AWS accounts from the same domain in the same region. If your DNS provider does not allow you to have multiple TXT records with the same name, there are two workarounds. The first workaround, if your DNS provider allows it, is to assign multiple values to the TXT record. For example, if your DNS is managed by Amazon Route 53, you can set up multiple values for the same TXT record as follows:
  1. In the Route 53 console, choose the _amazonses TXT record you added when you verified your domain in the first region.
  2. In the Value box, press Enter after the first value.
  3. Add the value for the additional region, and save the record set.

The other workaround is that if you only need to verify your domain twice, you can verify it once with _amazonses in the TXT record name and the other time you can omit _amazonses from the record name entirely. We recommend the previous solution as a best practice, however.
- Your email address is provided by a web-based email service you do not have control over—You cannot successfully verify a domain that you do not own. For example, if you want to send email through Amazon SES from a gmail address, you need to verify that email address specifically; you cannot verify gmail.com. For information about individual email address verification, see Verifying Email Addresses in Amazon SES (p. 43).
- Amazon SES reports that domain verification failed—You receive a "Domain Verification Failure" email from Amazon SES, and the domain displays a status of "failed" in the Domains tab of the Amazon SES console. This means that Amazon SES cannot find the necessary TXT record on your DNS server. Verify that the required TXT record is correctly published to your DNS server by using the procedure in How to Check Domain Verification Settings (p. 426), and look for the following possible errors:
  - Your DNS provider appended the domain name to the end of the TXT record—Adding a TXT record that already contains the domain name (such as _amazonses.example.com) may result in the duplication of the domain name (such as _amazonses.example.com.example.com). To avoid duplication of the domain name, add a period to the end of the domain name in the TXT record. This will indicate to your DNS provider that the record name is fully qualified (that is, no longer relative to the domain name), and prevent the DNS provider from appending an additional domain name.
You receive an email from Amazon SES that says your domain verification has been (or will be) revoked—Amazon SES can no longer find the required TXT record on your DNS server. The notification email will inform you of the length of time in which you must re-publish the TXT record before your domain verification status is revoked.

Note
You can review the required TXT record information in the Amazon SES console by using the following instructions. In the navigation pane, under Identities, choose Domains. In the list of domains, choose (not just expand) the domain to display the domain verification settings, which include the TXT record name and value.

If your domain verification status is revoked, you must restart the verification procedure in Verifying Domains in Amazon SES (p. 52) from the beginning, just as if the revoked domain were an entirely new domain. After you publish the TXT record to your DNS server, verify that the TXT record is correctly published by using How to Check Domain Verification Settings (p. 426).

How to Check Domain Verification Settings

You can check that your Amazon SES domain verification TXT record is published correctly to your DNS server by using the following procedure. This procedure uses the nslookup tool, which is available for Windows and Linux. On Linux, you can also use dig.

The commands in these instructions were executed on Windows 7, and the example domain we use is ses-example.com.

In this procedure, you first find the DNS servers that serve your domain, and then query those servers to view the TXT records. You query the DNS servers that serve your domain because those servers contain the most up-to-date information for your domain, which can take time to propagate to other DNS servers.

To verify that your domain verification TXT record is published to your DNS server

1. Find the name servers for your domain by taking the following steps.
   a. Go to the command line. To get to the command line on Windows 7, choose Start and then type cmd. On Linux-based operating systems, open a terminal window.
   b. At the command prompt, type the following, where <domain> is your domain. This will list all of the name servers that serve your domain.

      nslookup -type=NS <domain>

      If your domain was ses-example.com, this command would look like:

      nslookup -type=NS ses-example.com

      The command's output will list the name servers that serve your domain. You will query one of these servers in the next step.

2. Verify that the TXT record is correctly published by taking the following steps.
   a. At the command prompt, type the following, where <domain> is your domain, and <name server> is one of the name servers you found in step 1.

      nslookup -type=TXT _amazonses.<domain> <name server>
In our *ses-example.com* example, if a name server that we found in step 1 was called *ns1.name-server.net*, we would type the following:

```
nslookup -type=TXT _amazonses.ses-example.com ns1.name-server.net
```

b. In the output of the command, verify that the string that follows `text =` matches the TXT value you see when you choose the domain in the Identities list of the Amazon SES console.

In our example, we are looking for a TXT record under `_amazonses.ses-example.com` with a value of `fmxqxT/icOYx4aA/bEUrDPMeax9/s3frblS+niixmqk=`. If the record is correctly published, we would expect the command to have the following output:

```
_amazonses.ses-example.com text = "fmxqxT/icOYx4aA/bEUrDPMeax9/s3frblS+niixmqk="
```

**Amazon SES DKIM Problems**

If you attempt to set up Easy DKIM using the procedure in *Easy DKIM in Amazon SES (p. 117)* and you encounter problems, review the possible causes and solutions below.

- **You set up Easy DKIM successfully, but your messages are not being DKIM-signed**—Possible problems are:
  - Make sure that Easy DKIM is enabled for the appropriate identity. To enable Easy DKIM for an identity in the Amazon SES console, choose the email address or domain in the Identities list. On the Details page for the email address or domain, expand DKIM, and then choose Enable to enable DKIM.
  - You could be sending from an individually verified email address that does not have DKIM-signing enabled. If you set up Easy DKIM for a domain, it will apply to all email addresses in that domain except for email addresses that you individually verified. Individually verified email addresses use separate settings. If this is your issue, either remove the email address from your identity list (its settings will then be inherited from the verified domain's settings) or enable Easy DKIM for the email address as explained above.
  - If you are using Amazon SES in multiple regions or with multiple AWS accounts, you must perform the Easy DKIM set-up procedure described in *Easy DKIM in Amazon SES (p. 117)* for each region and account for which you want to use Easy DKIM. Amazon SES will generate a unique set of DNS records for each domain/account/region combination. You will need to add all of these records to your DNS server. If you remove the necessary DNS records for a specific region or account, Amazon SES will disable DKIM signing only for that account in that region, and notify you by email so that you can take action.

- **Your domain's DKIM details in the Amazon SES console show DKIM: waiting on sender verification...**
  - **DKIM Verification Status: pending verification**—Your DKIM status is pending, which means that Amazon SES has not yet detected the required CNAME records on your DNS server, which you should have published during the Easy DKIM set-up procedure (*Easy DKIM in Amazon SES (p. 117)*). If your DKIM status is pending, see the following articles on the Amazon SES blog:
    - **DKIM Troubleshooting Series: Your DKIM Status is Pending**
    - **DKIM Troubleshooting Series: Your DKIM Status is Still Pending**

- **When queried, your DNS servers successfully return the Amazon SES DKIM CNAME records, but return SERVFAIL for the TXT records**—Your DNS provider might have problems redirecting CNAME records. Note that Amazon SES and ISPs query for TXT records. To comply with the DKIM specification, your DNS servers must be able to respond to TXT record queries as well as CNAME record queries. If your DNS provider cannot respond to TXT record queries, an alternative is to use Amazon Route 53 for your DNS hosting.
• **Your emails are being DKIM-signed, but the DKIM signature is not validating**—See [DKIM Troubleshooting Series: Why is My Signature Not Validating?](https://amzn.to/3iG0OuM) on the Amazon SES blog.

• **You receive an email from Amazon SES that says your DKIM setup has been (or will be) revoked**—This means that Amazon SES can no longer find the required CNAME records on your DNS server. The notification email will inform you of the length of time in which you must re-publish the CNAME records before your DKIM setup status is revoked and DKIM signing is disabled. If your DKIM setup is revoked, you must restart the DKIM set-up procedure in Easy DKIM in Amazon SES (p. 117) from the beginning.

• **You do not have DKIM-signing enabled, yet your message headers contain a DKIM signature**—The DKIM signature you are seeing contains `d=amazonses.com` and is automatically added by Amazon SES.

• **Your emails contain two DKIM signatures**—The extra DKIM signature, which contains `d=amazonses.com`, is automatically added by Amazon SES. You can ignore it.

### Amazon SES Notification Problems

If you encounter a problem with bounce, complaint, or delivery notifications, review the possible causes and solutions below.

• **You receive bounce notifications via Amazon SNS, but you don’t know which recipients the notifications correspond to**—In the future, to associate a bounce notification with a given recipient, you have the following options:
  - Since Amazon SES doesn't retain any custom message IDs that you have added, store a mapping between an identifier and the Amazon SES message ID that Amazon SES passes back to you when it accepts the email.
  - In each call to Amazon SES, send to a single recipient, rather than sending a single message to multiple recipients.
  - You can enable feedback forwarding via email, which will forward the full bounce message to you.

• **You receive complaint notifications via Amazon SNS or email feedback forwarding, but you don’t know which recipients the notifications correspond to**—Some ISPs redact the complained recipient's email address before passing the complaint notification to Amazon SES. To enable you to find the recipient's email address, your best option is to store your own mapping between an identifier and the Amazon SES message ID that Amazon SES passes back to you when it accepts the email. Note that Amazon SES does not retain any custom message IDs that you add.

• **You want to set up notifications to go to an Amazon SNS topic you don’t own**—The owner of that topic must configure an Amazon SNS access policy that allows your account to call the `SNS:Publish` action on their topic. For information about how to control access to your Amazon SNS topic through the use of IAM policies, see [Managing Access to Your Amazon SNS Topics](https://docs.aws.amazon.com/sns/latest/dg/using-iam-policy-structure-tutorial.html) in the *Amazon Simple Notification Service Developer Guide*.

### Removing an Email Address from the Amazon SES Suppression List

Amazon SES maintains a suppression list of recipient email addresses that have recently caused a hard bounce for any Amazon SES customer. If you try to send an email through Amazon SES to an address that is on the suppression list, the call to Amazon SES succeeds, but Amazon SES treats the email as a hard bounce instead of attempting to send it. Like any hard bounce, suppression list bounces count towards your sending quota and your bounce rate. An email address can remain on the suppression list for up to 14 days.

The only way you will know if an address is on the suppression list is that you will receive a suppression list bounce when you send to it. There is no way to query the suppression list in advance.
Important
As with any email address that hard bounces, you should remove addresses that cause a suppression list bounce from your mailing list unless you are absolutely sure the address is valid, because suppression list bounces count towards your bounce rate and a high bounce rate puts your account at risk of being shut down. If you remove an address from the suppression list when it is indeed undeliverable, then the next time you or another Amazon SES customer sends an email to that address, it will hard bounce and the address will go back on the suppression list.

If you are sure that an address on the suppression list is valid, you can remove it from the list by using the following procedure. Although each AWS region has a separate suppression list, if you remove an address from the suppression list of one region, the address is removed from the suppression list of all regions.

To remove an email address from the suppression list

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the Navigation pane, choose Suppression List Removal.
3. In the Email Address field, type the email address that you want to remove from the suppression list.
4. In the Type characters field, type the characters that you see in the image above it.
5. Choose Submit.

After you submit the form, you can fill out the form for another email address. Suppression list removal requests are processed immediately.

Increasing Throughput with Amazon SES

When you send emails, you can call Amazon SES as frequently as your maximum send rate allows. (For more information about your maximum send rate, see Managing Your Amazon SES Sending Limits (p. 127).) However, each call to Amazon SES takes time to complete.

If you are making multiple calls to Amazon SES using the Amazon SES API or the SMTP interface, you may want to consider the following tips to help you improve your throughput:

- **Measure your current performance to identify bottlenecks**—A possible performance test involves sending multiple test emails as quickly as possible within a code loop in your application. Measure the round-trip latency of each SendEmail request. Then, incrementally launch additional instances of the application on the same machine, and watch for any impact on network latency. You may also want to run this test on multiple machines and on different networks to help pinpoint any possible machine resource bottlenecks or network bottleneck that may exist.

- **(API only) Consider using persistent HTTP connections**—Rather than incurring the overhead of establishing a separate new HTTP connection for each API request, use persistent HTTP connections. That is, reuse the same HTTP connection for multiple API requests.

- **Consider using multiple threads**—When an application uses a single thread, the application code calls the Amazon SES API and then synchronously waits for an API response. Sending emails is typically an I/O-bound operation, and doing the work from multiple threads provides better throughput. You can send concurrently using as many threads of execution as you wish.

- **Consider using multiple processes**—Using multiple processes can help increase your throughput because you will have more concurrent active connections to Amazon SES. For example, you can segment your intended emails into multiple buckets, and then run multiple instances of your email sending script simultaneously.
• **Consider using a local mail relay**—Your application can quickly transmit messages to your local mail server, which can then help to buffer the messages and asynchronously transmit them to Amazon SES. Some mail servers support delivery concurrency, which means that even if your application is generating emails to the mail server in a single-threaded fashion, the mail server will use multiple threads when sending to Amazon SES. For more information, see Integrating Amazon SES with Your Existing Email Server (p. 81).

• **Consider hosting your application closer to the Amazon SES API endpoint**—You may wish to consider hosting your application in a data center close to the Amazon SES API endpoint, or on an Amazon EC2 instance in the same AWS Region as the Amazon SES API endpoint. This may help to decrease network latency between your application and Amazon SES, and improve throughput. For a list of Amazon SES endpoints, see Regions and Amazon SES (p. 409).

• **Consider using multiple machines**—Depending on the system configuration on your host machine, there may be a limit on the number of simultaneous HTTP connections to a single IP address, which may limit the benefits of parallelism once you exceed a certain number of concurrent connections on a single machine. If this is a bottleneck, you may wish to consider making concurrent Amazon SES requests using multiple machines.

• **Consider using the Amazon SES query API instead of the SMTP endpoint**—Using the Amazon SES query API enables you to submit the email send request using a single network call, whereas interfacing with the SMTP endpoint involves an SMTP conversation which consists of multiple network requests (for example, EHLO, MAIL FROM, RCPT TO, DATA, QUIT). For more information about the Amazon SES query API, see Using the Amazon SES API to Send Email (p. 99).

• **Use the Amazon SES mailbox simulator to test your maximum throughput**—To test any changes you may implement, you can use the mailbox simulator. The mailbox simulator can help you to determine your system’s maximum throughput without using up your daily sending quota. For information about the mailbox simulator, see Testing Amazon SES Email Sending (p. 162).

If you are accessing Amazon SES through its SMTP interface, see Amazon SES SMTP Issues (p. 430) for specific SMTP-related issues that may affect throughput.

---

**Amazon SES SMTP Issues**

If you are having problems sending email through the Amazon SES Simple Mail Transfer Protocol (SMTP) interface, review the possible causes and solutions below. For general information about sending email through the Amazon SES SMTP interface, see Using the Amazon SES SMTP Interface to Send Email (p. 66).

• **You are unable to connect to the Amazon SES SMTP endpoint.**

Problems connecting to the Amazon SES SMTP endpoint are most commonly related to the following issues:

• **Incorrect credentials** — The credentials that you use to connect to the SMTP endpoint are different from your AWS credentials. To obtain your SMTP credentials, see Obtaining Your Amazon SES SMTP Credentials (p. 67). For more information about credentials, see Using Credentials With Amazon SES (p. 369).

• **Network or firewall issues** — Your network might be blocking outbound connections over the port you’re trying to send email from. To determine if an issue on your local network is causing connection issues, type the following command at the command line, replacing `port` with the port you’re trying to use (typically 25, 465, 587, 2465, or 2587): `telnet email-smtp.us-west-2.amazonaws.com port`

If you are able to connect to the SMTP server using this command, and you are trying to connect to Amazon SES using TLS Wrapper or STARTTLS, complete the procedures shown in Using the Command Line to Send Email Through the Amazon SES SMTP Interface (p. 95).
If you cannot connect to the Amazon SES SMTP endpoint using telnet or openssl, it indicates that something in your network (such as a firewall) is blocking outbound connections over the port you’re trying to use. Work with your network administrator to diagnose and fix the problem.

- **You are sending to Amazon SES from an Amazon EC2 instance via port 25 and you cannot reach your Amazon SES sending limits or you are receiving time outs.**

  Amazon EC2 imposes default sending limits on email sent via port 25 and throttles outbound connections if you attempt to exceed those limits. To remove these limits, submit an Amazon EC2 Request to Remove Email Sending Limitations. You can also connect to Amazon SES using ports 465 or 587, neither of which is throttled.

- **Network errors are causing dropped emails.**

  Ensure that your application uses retry logic when it connects to the Amazon SES SMTP endpoint, and that your application can detect and retry message delivery in case of a network error. SMTP is a verbose protocol, and sending an email using this protocol requires several network round trips. Because of the nature of SMTP, the potential for network errors increases.

- **You lose connection with the SMTP endpoint.**

  Lost connections are most commonly caused by the following issues:

  - **MTU size** – If you receive a time-out error message, the Maximum Transmission Unit (MTU) of the network interface for the computer you're using to connect to the Amazon SES SMTP interface may be too large. To resolve this issue, set the MTU size on that computer to 1500 bytes.

    For more information about setting the MTU size on Windows, Linux, and macOS operating systems, see Queries Appear to Hang in the Client and Do Not Reach the Cluster in the Amazon Redshift Cluster Management Guide.

    For more information about setting the MTU size for an Amazon EC2 instance, see Network Maximum Transmission Unit (MTU) for Your EC2 Instance in the Amazon EC2 User Guide for Linux Instances.

  - **Long-lived connections** – The Amazon SES SMTP endpoint runs on a fleet of Amazon EC2 instances behind an Elastic Load Balancer (ELB). In order to ensure that the system is up-to-date and fault tolerant, active Amazon EC2 instances are periodically terminated and replaced with new instances. Because your application connects to an Amazon EC2 instance through the ELB, the connection becomes invalid when the Amazon EC2 instance is terminated. You should establish a new SMTP connection after you have delivered a fixed number of messages via a single SMTP connection, or if the SMTP connection has been active for some amount of time. You will need to experiment to find appropriate thresholds depending on where your application is hosted and how it submits email to Amazon SES.

- **You want to know the IP addresses of the Amazon SES SMTP mail servers so that you can whitelist the IP addresses with your network.**

  The IP addresses for the Amazon SES SMTP endpoints reside behind load balancers and the IP addresses change frequently. For this reason, we are not able to provide a definitive list of IP addresses for the Amazon SES endpoints. We recommend that you only whitelist based on DNS and not static IP addresses.

- **You are integrating Amazon SES with Sendmail or Postfix, and your server cannot authenticate because the hostname does not match.**

  If you completed the procedures in Integrating Amazon SES with Your Existing Email Server (p. 81), but you receive an authentication error stating that the hostname does not match, complete the appropriate troubleshooting steps for your mail server:

  - **If you use Sendmail** – After you complete step 2 of Integrating Amazon SES with Sendmail (p. 85), add one of the following lines to `/etc/mail/authinfo`, replacing `USERNAME` and `PASSWORD` with your SMTP user name and password.
### SMTP Issues

<table>
<thead>
<tr>
<th>Region</th>
<th>Content to Add</th>
</tr>
</thead>
</table>

After you complete step 5 of [Integrating Amazon SES with Sendmail (p. 85)](#), add one of the following lines to `/etc/mail/access`:

<table>
<thead>
<tr>
<th>Region</th>
<th>Content to Add</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (N. Virginia)</td>
<td>Connect:ses-smtp-prod-335357831.us-east-1.elb.amazonaws.com RELAY</td>
</tr>
<tr>
<td>EU (Ireland)</td>
<td>Connect:ses-smtp-eu-west-1-prod-345515633.eu-west-1.elb.amazonaws.com RELAY</td>
</tr>
</tbody>
</table>

- **If you use Postfix** – In part 2 of [Integrating Amazon SES with Postfix (p. 82)](#), before you encrypt the file, add one of the following lines to `/etc/postfix/sasl_passwd`, replacing `USERNAME` and `PASSWORD` with your SMTP user name and password.

<table>
<thead>
<tr>
<th>Region</th>
<th>Content to Add</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (N. Virginia)</td>
<td>ses-smtp-prod-335357831.us-east-1.elb.amazonaws.com:25 <code>USERNAME:PASSWORD</code></td>
</tr>
<tr>
<td>EU (Ireland)</td>
<td>ses-smtp-eu-west-1-prod-345515633.eu-west-1.elb.amazonaws.com:25 <code>USERNAME:PASSWORD</code></td>
</tr>
</tbody>
</table>
SMTP Response Codes Returned by Amazon SES

This topic contains a list of SMTP response codes that are returned by Amazon SES. The way in which errors are handled depends on the SMTP client that you use; some SMTP clients may not display error codes at all.

You should retry SMTP requests that receive 4xx errors. In this case, to reduce the likelihood of generating duplicates, we recommend that you implement an exponential retry method with progressively longer waits (5, 10, and 30 seconds) between consecutive timeouts. If the third retry call does not succeed, perform another set of retries after 20 minutes. For an example implementation that uses an exponential retry policy with Amazon SES, see How to handle a "Throttling - Maximum sending rate exceeded" error on the Amazon SES blog.

**Note**
AWS SDKs implement retry logic automatically, although they use the HTTPS interface instead of SMTP.

SMTP client errors (5xx) indicate that you need to revise the request to correct the problem before trying again. For example, if your AWS authentication credentials are invalid, you must update your setup to use the proper credentials before trying to send the email again.

<table>
<thead>
<tr>
<th>Description</th>
<th>Response code</th>
<th>More information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication successful</td>
<td>235 Authentication successful</td>
<td>N/A</td>
</tr>
<tr>
<td>Successful delivery</td>
<td>250 Ok <code>&lt;Message ID&gt;</code></td>
<td><code>&lt;Message ID&gt;</code> is a string of characters that Amazon SES uses to uniquely identify a message.</td>
</tr>
<tr>
<td>Daily sending quota exceeded</td>
<td>454 Throttling failure: Daily message quota exceeded</td>
<td>You have exceeded the maximum number of emails that Amazon SES permits you to send in a 24-hour period. For more information, see Managing Your Amazon SES Sending Limits (p. 127).</td>
</tr>
<tr>
<td>Maximum send rate exceeded</td>
<td>454 Throttling failure: Maximum sending rate exceeded</td>
<td>You have exceeded the maximum number of emails that Amazon SES permits you to send per second. For more information, see Managing Your Amazon SES Sending Limits (p. 127).</td>
</tr>
<tr>
<td>Amazon SES issue when validating SMTP credentials</td>
<td>454 Temporary authentication failure</td>
<td>Possible reasons include, but are not limited to:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• There is a problem with the encryption between your email-sending application and Amazon SES. Note that you need to use an encrypted connection when you connect to Amazon SES. For more information, see Connecting to the Amazon SES SMTP Endpoint (p. 70).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Amazon SES could be experiencing an issue. Check the</td>
</tr>
<tr>
<td>Description</td>
<td>Response code</td>
<td>More information</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Problem receiving the request</td>
<td>454 Temporary service failure</td>
<td>Amazon SES did not successfully receive the request and therefore did not send the message. Please retry the request.</td>
</tr>
<tr>
<td>Incorrect credentials</td>
<td>530 Authentication required</td>
<td>Your email-sending application did not attempt to authenticate with Amazon SES when it tried to connect to the Amazon SES SMTP interface. For an example of how to set up an email-sending application to authenticate with Amazon SES, see Configuring Email Clients to Send Through Amazon SES (p. 71).</td>
</tr>
<tr>
<td>Authentication Credentials Invalid</td>
<td>535 Authentication Credentials Invalid</td>
<td>Your email-sending application did not provide the correct SMTP credentials to Amazon SES. Note that your SMTP credentials are not the same as your AWS credentials. For more information, see Obtaining Your Amazon SES SMTP Credentials (p. 67).</td>
</tr>
<tr>
<td>Account not subscribed to Amazon SES</td>
<td>535 Account not subscribed to SES</td>
<td>The AWS account that owns the SMTP credentials is not signed up for Amazon SES.</td>
</tr>
<tr>
<td>User not authorized to call the Amazon SES SMTP endpoint</td>
<td>554 Access denied: User &lt;User ARN&gt; is not authorized to perform ses:SendRawEmail on resource &lt;Identity ARN&gt;</td>
<td>The AWS Identity and Access Management (IAM) policy or the Amazon SES sending authorization policy of the user who owns the SMTP credentials is not allowed to call the Amazon SES SMTP endpoint. For information about how to get SMTP credentials for an existing IAM user, see Obtaining Amazon SES SMTP Credentials by Converting AWS Credentials (p. 68).</td>
</tr>
<tr>
<td>Description</td>
<td>Response code</td>
<td>More information</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Unverified email address</td>
<td>554 Message rejected: Email address is not verified. The following identities failed the check in region <code>&lt;region&gt;</code>: <code>&lt;identity1&gt;</code>, <code>&lt;identity2&gt;</code>, <code>&lt;identity3&gt;</code></td>
<td>You are trying to send email from an email address or domain that you have not verified with Amazon SES (p. 43). This error could apply to the “From”, “Source”, “Sender”, or “Return-Path” address. If your account is still in the sandbox, you also must verify every recipient email address except for the recipients provided by the Amazon SES mailbox simulator (p. 162). If Amazon SES is not able to show all of the failed identities, the error message ends with an ellipsis. <strong>Note</strong> Amazon SES has endpoints in multiple AWS regions (p. 409), and email address verification status is separate for each AWS region. You must complete the verification process for each sender in the AWS region(s) you want to use.</td>
</tr>
</tbody>
</table>
Amazon SES Frequently Asked Questions (FAQs)

This section contains answers to several frequently asked questions related to using Amazon SES.

This section contains FAQs for the following topics:

- Enforcement FAQs (p. 436)
- IP Blacklist FAQs (p. 449)
- Email Metrics FAQs (p. 450)

Amazon SES Enforcement FAQs

If the emails you send result in excessive bounces, complaints, or other issues, your sending abilities might be placed on probation or suspended. This process is called enforcement. In these cases, you will receive a notification at the email address associated with your AWS account.

This section contains FAQs on the following enforcement-related topics:

- Probations (p. 436)
- Suspensions (p. 438)
- Bounces (p. 440)
- Complaints (p. 442)
- Spamtraps (p. 446)
- Manual Investigations (p. 448)

Amazon SES Probation FAQ

Q1. I received a probation notice. What does that mean?

We have detected a significant issue with the sending on your account and we're giving you time to fix it. You can still send normally for now, but if you don't fix the problem in the allotted timeframe or sending allowance, your Amazon SES sending privileges might be suspended.

Q2. Will I always be notified if I am put on probation?

Yes. You will receive a notification at the email address of the AWS account associated with the Amazon SES probation.

Q3. Will the Amazon SES probation affect my use of other AWS services?

You will still be able to use other AWS services while your Amazon SES account is on probation. However, if you request a service limit increase for another AWS service that sends outbound communications (such as Amazon SNS), that request may be denied until the probation on your Amazon SES account is lifted.
Q4. What should I do if I'm on probation?

You should do the following:

- If your situation allows it, stop sending mail until you fix the problem. Although probation does not affect your ability to send mail through Amazon SES, if you continue to send mail without first making changes, you are putting your continued sending at risk.
- Look at the email you received from us for a summary of the issue.
- Investigate your sending to determine what aspect of your sending specifically triggered the issue.
- Once you have made your fixes, send us an appeal telling us about the fixes you made (see Q6. How do I submit an appeal? (p. 437)). Note that you should appeal your probation only after you’ve made your changes—do not submit an appeal outlining changes you plan to make. If you do, we will ask you to contact us again once the fixes are actually in place. If we find that you have fixed the problem, we’ll take you off probation.
- Be sure to provide any information we specifically request. We need this information to evaluate your case.

Q5. What is an appeal?

An appeal is when you reply to a probation or suspension notification (or email ses-enforcement@amazon.com from the email address associated with your AWS account) and provide the specific information we need to determine whether we can remove the probation or suspension. For a list of information to provide, see Q6. How do I submit an appeal? (p. 437).

Q6. How do I submit an appeal?

Just reply to the probation notification. If you cannot find the probation notification, send your appeal to ses-enforcement@amazon.com from the email address associated with your AWS account. In your appeal, you should explain in as much detail as possible the following three things:

- An explanation of how and why you think the problem occurred.
- A list of changes you have already made to address the issue (not changes you plan to make).
- An explanation of why you believe these changes will prevent the problem from happening again.

Please read the FAQ specific to your issue (for example, bounces) to see if there is additional information you need to provide in your appeal.

**Note**

Failure to provide this information will delay the appeal process because we will request the remaining information before we can make a decision. In addition, be sure to provide any additional information we specifically request during the appeal correspondence.

Q7. What if my appeal isn't accepted?

We will respond to you explaining why your appeal wasn't accepted, and you will often have the option of appealing again after you address the issue. For example, we might ask you for more information, and once you provide the information, your appeal might be accepted. As another example, if you tell us how you will fix the problem and haven't actually fixed it, we'll ask you to contact us again once you've actually fixed the issue.

Q8. Can you help me diagnose the problem?

Typically we can give you only a high-level overview of your issue (for example, that you have a problem with bounces). You will need to investigate the root cause on your end.
Q9. How will I know if I'm off probation?

We will provide this information in our response to your appeal, or in some cases you will receive an automated notification at the email address associated with your AWS account. The notification will indicate either that you're off probation, or that your account has been suspended because you haven't fixed the problem.

Q10. Will I always have a probation period if there's a problem?

No. There are two cases in which you might not be provided a probation period:

- If your sending problem is very severe, you might be immediately suspended. As with any suspension, we will send you a notification at that time.
- If you show a pattern of repeated probations, eventually you might be suspended rather than being put on probation again. For this reason, it is important to address the underlying problem rather than just the specific incident that caused a specific probation. For instance, if a particular campaign triggers a probation, you must do more than simply stop that campaign. You need to determine which properties of the campaign were problematic and ensure that you have processes in place so that your future campaigns won't have the same issue.

Q11. What if I make my fixes shortly before the probation is due to expire?

Contact us through the appeal process to let us know that you fixed the problem.

Q12. Can I get help from my AWS representative or Premium Support?

If you are actively working with an AWS account representative, we will contact him or her regarding your probation, and he or she might be able to help you to better understand the problem. Feel free to contact your representative directly as well. If you are signed up for Premium Support, you might also want to engage that team to help with this process.

Amazon SES Suspension FAQ

Q1. I received a suspension notice. What does that mean?

We had to shut down your account due to a critical issue with your sending, so you can no longer send emails. There are three scenarios in which a suspension can occur:

- You are put on probation for a sending problem (for example, bounces), the probation expires, and the issue has not been resolved.
- Your problem is so severe that you were immediately suspended without a probation period.
- You have a history of repeated probations for a particular issue, and the issue reoccurred.

Q2. Will I always be notified if I am suspended?

Yes. You will receive a notification at the email address of the AWS account associated with the Amazon SES suspension.
Q3. Will the Amazon SES suspension affect my use of other AWS services?

You will still be able to use other AWS services while your Amazon SES account is suspended. However, if you request a service limit increase for another AWS service that sends outbound communications (such as Amazon SNS), that request may be denied until the suspension on your Amazon SES account is lifted.

Q4. What should I do if my account has been suspended?

You should do the following:

• Look at the email you received from us for a summary of the issue.
• Investigate your sending to determine what aspect of your sending specifically triggered the issue.
• Once you have fixed the issue, send us an appeal telling us about the fixes you made (see Q6. How do I submit an appeal? (p. 439)). Note that you should appeal your suspension only after you've made your changes—do not submit an appeal outlining changes you plan to make. If you do, we will ask you to contact us again once the fixes are actually in place.
• Be sure to provide any information we specifically request. We need this information to evaluate your case.

Q5. What is an appeal?

An appeal is when you reply to a probation or suspension notification (or email ses-enforcement@amazon.com from the email address associated with your AWS account) and provide the specific information we need to determine whether we can remove the probation or suspension. For a list of information to provide, see Q6. How do I submit an appeal? (p. 439).

Q6. How do I submit an appeal?

Just reply to the suspension notification. If you cannot find the suspension notification, send your appeal to ses-enforcement@amazon.com from the email address associated with your AWS account. In your appeal, you should explain in as much detail as possible the following three things:

• An explanation of how and why you think the problem occurred.
• A list of changes you have already made to address the issue (not changes you plan to make).
• An explanation of why you believe these changes will prevent the problem from happening again.

Read the FAQ specific to your issue (for example, bounces) to see if there is additional information you need to provide in your appeal.

**Note**

Failure to provide this information will delay the appeal process because we will request the remaining information before we can make a decision. In addition, be sure to provide any additional information we specifically request during the appeal correspondence.

Q7. What if my appeal isn't accepted?

We will respond to you explaining why your appeal wasn't accepted, and you will often have the option of appealing again after you address the issue. For example, we might ask you for more information, and once you provide the information, your appeal might be accepted. As another example, if you tell us how you will fix the problem and haven't actually fixed it, we'll ask you to contact us again once you've actually fixed the issue.
Q8. Can you help me diagnose the problem?

Typically we can give you only a high-level overview of your issue (for example, that you have a problem with bounces). You will need to investigate the root cause on your end.

Q9. How will I know if my account has been reinstated?

We will provide this information in our response to your appeal, or in some cases you will receive an automated notification at the email address associated with your AWS account. You can also try sending an email to yourself through Amazon SES (for example, using the Amazon SES console). If the attempt is successful, then you have been reinstated.

Q10. Can I get help from my AWS representative or Premium Support?

If you are actively working with an AWS account representative, we will contact him or her regarding your probation, and he or she might be able to help you to better understand the problem. Feel free to contact your representative directly as well. If you are signed up for Premium Support, you might also want to engage that team to help with this process.

Amazon SES Bounce FAQ

Q1. Why do you care about my bounces?

High bounce rates are often used by entities such as ISPs, mailbox providers, and anti-spam organizations as indicators that senders are engaging in low-quality email-sending practices and their email should be blocked or sent to the spam folder.

Q2. What should I do if I receive a probation or suspension notice for my bounce rate?

Fix the underlying problem and appeal to get your case reevaluated. For information about the appeal process, see the FAQs on probation and suspension. In your appeal, in addition to the information requested in the probation and suspension FAQs, tell us the following:

- The method you use to track your bounces
- How you ensure that the email addresses of new recipients are valid prior to sending to them.
  For example, which of the recommendations are you following in Q11. What can I do to minimize bounces? (p. 441)

Q3. What types of bounces count toward my bounce rate?

Your bounce rate includes only hard bounces to domains you have not verified. Hard bounces are permanent delivery failures such as "address does not exist." Temporary and intermittent failures such as "mailbox full," or bounces due to blocked IP addresses, do not count toward your bounce rate.

Q4. Do you disclose the Amazon SES bounce rate limits that trigger probation and suspension?

No, but you can find general bounce rate guidelines and tips on how to avoid bounces in the Amazon Simple Email Service Email Sending Best Practices whitepaper.
Q5. Over what period of time is my bounce rate calculated?

We don't calculate your bounce rate based on a fixed period of time, because different senders send at different rates. Instead, we look at what is called a representative volume, which represents a reasonable amount of mail with which to evaluate your sending practices. To be fair to both high-volume and small-volume senders, the representative volume is different for each user and changes as the user's sending patterns change.

Q6. Can I calculate my own bounce rate by using the information from the Amazon SES console or the GetSendStatistics API?

No. The bounce rate is calculated using representative volume (see Q5. Over what period of time is my bounce rate calculated? (p. 441)). Depending on your sending rate, your bounce rate can stretch farther back in time than the Amazon SES console or GetSendStatistics can retrieve. In addition, only emails to non-verified domains are considered when calculating your bounce rate. However, if you regularly monitor your bounce rates using those methods, you should still have a good indicator that you can use to catch problems before they get to levels that trigger a probation or suspension.

Q7. How can I find out which email addresses bounced?

Examine the bounce notifications that Amazon SES sends you. The email address to which Amazon SES forwards the notifications depends on how you sent the original messages, as described at Amazon SES Notifications Through Email (p. 227). You can also set up bounce notifications through Amazon Simple Notification Service (Amazon SNS), as described at Monitoring Using Amazon SES Notifications (p. 226). Note that simply removing bounced addresses from your list without any additional investigation might not solve the underlying problem. For information about what you can do to reduce bounces, see Q11. What can I do to minimize bounces? (p. 441).

Q8. If I haven't been monitoring my bounces, can you give me a list of addresses that have bounced?

No. We cannot give you a comprehensive list. You should be regularly monitoring your bounces by email or through Amazon SNS.

Q9. How should I handle bounces?

You need to remove bounced addresses from your mailing list and stop sending mail to them immediately. If you are a small sender, it might be sufficient to simply monitor bounces through email and manually remove bounced addresses from your mailing list. If your volume is higher, you will probably want to set up automation for this process, either by programatically processing the mailbox where you receive bounces, or by setting up bounce notifications through Amazon SNS. For more information, see Monitoring Using Amazon SES Notifications (p. 226).

Q10. Could my emails be bouncing because I've reached my sending limits?

No. Bounces have nothing to do with sending limits. If you try to exceed your sending limits, you will receive an error from the Amazon SES API or SMTP interface when you try to send an email.

Q11. What can I do to minimize bounces?

First, be sure that you are aware of your bounces (see Q7. How can I find out which email addresses bounced? (p. 441)). Then follow these guidelines:
• Do not buy, rent, or share email addresses. Use only addresses that specifically requested your mail.
• Remove bounced email addresses from your list.
• When you ask for email addresses, ask twice and require a match to minimize typos.
• Use double opt-in to sign up new users. That is, when a new user signs up, send them a confirmation email that they need to click before receiving any additional mail. This prevents people from signing up other people as well as accidental signups.
• If you must send to addresses that you haven't mailed lately (and thus you can't be confident that the addresses are still valid), do so only with a small portion of your overall sending. For more information, see our blog post Never send to old addresses, but what if you have to?.
• Ensure that you are not structuring signups to encourage people to use fictional addresses. Either do not provide any value to new users until their email address is verified, or ask for their address only when they specifically sign up to receive mail.
• If you have an "email a friend" feature, be sure you use CAPTCHA or a similar mechanism to discourage automated use of the feature, and do not allow arbitrary content to be inserted by the user. For more information about CAPTCHA, see http://www.captcha.net/.
• If you are using Amazon SES for system notifications, ensure that you are sending the notifications to real addresses that can receive mail. Also consider turning off notifications that you do not need.
• If you are testing a new system, be sure you are either sending to real addresses that can receive email, or you are using the Amazon SES mailbox simulator. For more information, see Testing Amazon SES Email Sending (p. 162).

Amazon SES Complaint FAQ

Q1. What is a complaint?
A complaint occurs when a recipient reports that they do not want to receive an email. They might have clicked the "This is spam" button in their email client, complained to their email provider, notified Amazon SES directly, or through some other method. This topic includes general information about complaints. If your notification contains specific information about the source of the complaints, also read the relevant topic: Amazon SES Complaints Through ISP Feedback Loops FAQ (p. 443), Amazon SES Complaints Directly from Recipients FAQ (p. 444), or Amazon SES Complaints Through Email Providers FAQ (p. 445).

Q2. Why do you care about my complaints?
High complaint rates are often used by entities such as ISPs, email providers, and anti-spam organizations as indicators that a sender is sending to recipients who did not specifically sign up to receive emails, or that the sender is sending content that is different from the type that recipients signed up for.

Q3. What should I do if I receive a probation or suspension notice for my complaint rate?
Review your list acquisition process and the content of your emails to try to understand why your recipients might not appreciate your email. Once you have determined the cause, fix the underlying problem and appeal to get your case reevaluated. For information about the appeal process, see the FAQs on probation and suspension.

Q4. What can I do to minimize complaints?
First, be sure that you monitor the complaints that Amazon SES can notify you about, which are complaints that Amazon SES receives through ISP feedback loops (see the Amazon SES Complaints Through ISP Feedback Loops FAQ (p. 443)). Then follow these guidelines:
• Do not buy, rent, or share email addresses. Use only addresses that specifically requested your mail.
• Use double opt-in to sign up new users. That is, when users sign up, send them a confirmation email that they need to click before receiving any additional mail. This prevents people from signing up other people as well as accidental signups.
• Monitor engagement with the mail you send and stop sending to recipients who do not open or click your messages.
• When new users sign up, be clear about the type of email they will receive from you, and ensure that you send only the type of mail that they signed up for. For example, if users sign up for news updates, do not send them advertisements.
• Ensure that your mail is well-formatted and looks professional.
• Ensure that your mail is clearly from you and cannot be confused for something else.
• Provide users an obvious and easy way to unsubscribe from your mail.

Amazon SES Complaints Through ISP Feedback Loops FAQ

This topic provides information about complaints that Amazon SES receives through feedback loops. For general information that applies to all types of complaints, see the Amazon SES Complaint FAQ (p. 442).

Q1. How is this type of complaint reported?

Most email client programs provide a button labeled "Mark as Spam," or similar, which moves the message to a spam folder and forwards it to the ISP. Additionally, most ISPs maintain an abuse address (e.g., abuse@example.net), where users can forward unwanted emails and request that the ISP take action to prevent them. If the Amazon SES has a feedback loop (FBL) set up with the ISP, then the ISP will send the complaint back to Amazon SES.

Q2. Are these complaints included in the complaint rate statistic shown in the Amazon SES console and returned by the GetSendStatistics API?

Yes. Note, however, that the complaint rate statistic does not include complaints from ISPs that do not provide feedback to Amazon SES. Nevertheless, the complaint rate from domains that provide feedback is likely to be representative of the rest of your sending as well.

Q3. How can I be notified of these complaints?

You can be notified through email or through Amazon SNS notifications. See the set-up instructions in Monitoring Using Amazon SES Notifications (p. 226).

Q4. What should I do if I receive a complaint notification through email or through Amazon SNS?

First, you need to remove addresses that generated complaints from your mailing list and stop sending mail to them immediately. Do not even send an email that says you have received the request to unsubscribe. You will probably want to set up automation for this process, either by programmatically processing the mailbox where you receive complaints, or by setting up complaint notifications through Amazon SNS. For more information, see Monitoring Using Amazon SES Notifications (p. 226).

Then, take a close look at your sending to determine why your recipients do not appreciate the mail you are sending, and address that underlying problem. For every person who complains, there are potentially dozens who didn't appreciate your mail who did not (or were not able to) complain. If all you do is remove the recipients who actually complain, you are not addressing the underlying problem with your sending.
Q5. Do you disclose the Amazon SES complaint rate limits that trigger probation and suspension?

No, but you can find general complaint rate guidelines and tips on how to avoid complaints in the Amazon Simple Email Service Email Sending Best Practices whitepaper.

Q6. Over what period of time is my complaint rate calculated?

We don't calculate your complaint rate based on a fixed period of time, because different senders send at different rates. Instead, we look at what is called a *representative volume*, which represents a reasonable amount of mail with which to evaluate your sending practices. To be fair to both high-volume and small-volume senders, the representative volume is different for each user and changes as the user's sending patterns change. Additionally, the complaint rate isn't calculated based on every email. It is calculated as the percentage of complaints on mail sent to domains that send complaint feedback to Amazon SES.

Q7. Can I calculate my own complaint rate by using metrics from the Amazon SES console or the GetSendStatistics API?

No. There are two primary reasons for this:

- The complaint rate is calculated using representative volume (see Q6). Depending on your sending rate, your complaint rate can stretch farther back in time than the Amazon SES console or GetSendStatistics can retrieve. However, if you regularly monitor your complaint rates using those methods, you should still have a good indicator that you can use to catch problems before they get to levels that trigger a probation or suspension.
- When calculating complaint rate, not every email counts. Complaint rate is calculated as the percentage of complaints on mail sent to domains that send complaint feedback to Amazon SES.

Q8. How can I find out which email addresses complained?

Examine the complaint notifications that Amazon SES sends you through email or through Amazon SNS (see Monitoring Using Amazon SES Notifications (p. 226)). However, different ISPs provide differing amounts of information, and some ISPs redact the complained recipient's email address before passing the complaint notification to Amazon SES. To enable you to find the recipient's email address in the future, your best option is to store your own mapping between an identifier and the Amazon SES message ID that Amazon SES passes back to you when it accepts the email. Note that Amazon SES does not retain any custom message IDs that you add.

Q9. If I haven't been monitoring my complaints, can you give me a list of addresses that have complained?

Unfortunately, we can't give you a comprehensive list. However, you can monitor future complaints by email or through Amazon SNS.

Q10. Can I get a sample email?

We cannot send you a sample email upon request, but you might find this information in the complaint notification. See the answer to Q8.

Amazon SES Complaints Directly from Recipients FAQ

This topic provides information about complaints that Amazon SES receives directly from recipients. For general information that applies to all types of complaints, see the Amazon SES Complaint FAQ (p. 442).
Q1. How is this type of complaint reported?

Multiple recipients directly contacted Amazon SES about your mail through email or some other means.

Q2. Are these complaints included in the complaint rate statistic shown in the Amazon SES console and returned by the GetSendStatistics API?

No. The complaint rate statistic you retrieve using the Amazon SES console or the GetSendStatistics API only includes complaints that Amazon SES receives through ISP feedback loops. For more information about those types of complaints, see the Amazon SES Complaints Through ISP Feedback Loops FAQ (p. 443).

Q3. Why haven't I heard about these complaints through email feedback notifications or through Amazon SNS?

Email feedback forwarding and Amazon SNS notifications only include complaints that Amazon SES receives through ISP feedback loops. You will not receive notifications for complaints that recipients filed directly with Amazon SES.

Q4. How can I find out which email addresses complained?

We are unable to disclose the addresses of recipients who complained, but we can say that it is more than one recipient, and the number of complaints is significant when taking your current sending volume into consideration. However, rather than focusing on removing individual recipients from your list, you need to focus on finding and fixing the underlying problem. Start by reviewing your list acquisition process and the content of your emails to try to understand why your recipients might not appreciate your email.

Q5. Can I get a sample email?

Unfortunately, we are unable to provide an example of an email that triggered a recipient to directly complain.

Q6. I have received a probation notice for direct recipient complaints. What should I do?

As soon as possible, fix your system so that your mailing list only includes recipients who have specifically signed up to receive your mail, and ensure you are sending content that your recipients actually want. Then, please email us with the details of your changes so that we can start the process of re-evaluating your case. If three weeks pass and we don't hear from you at all, we will have to disable your sending if we are still getting complaints about your mail.

Amazon SES Complaints Through Email Providers FAQ

This topic provides information about complaints that Amazon SES receives through email providers (also called mailbox providers). For general information that applies to all types of complaints, see the Amazon SES Complaint FAQ (p. 442).

Q1. How is this type of complaint reported?

An email provider reported to Amazon SES that a significant number of its customers marked your emails as spam. The report was provided to Amazon SES through a means other than the feedback loops described in the Amazon SES Complaints Through ISP Feedback Loops FAQ (p. 443).
Q2. Are these complaints included in the complaint rate statistic shown in the Amazon SES console and returned by the GetSendStatistics API?

No. The complaint rate statistic you retrieve using the Amazon SES console or the GetSendStatistics API only includes complaints that Amazon SES receives through ISP feedback loops.

Q3. Why haven't I heard about these complaints through email feedback notifications or through Amazon SNS?

Email feedback forwarding and Amazon SNS notifications only include complaints that Amazon SES receives through ISP feedback loops.

Q4. How can I find out which email addresses complained?

Email providers typically do not disclose this information. However, rather than focusing on removing individual recipients from your list, you need to focus on finding and fixing the underlying problem. Start by reviewing your list acquisition process and the content of your emails to try to understand why your recipients might not appreciate your email.

Q5. Can I get a sample email?

No. Email providers typically do not provide an example email.

Q6. I have received a probation or shutdown notice for this type of complaint. What should I do?

Fix your system so that your mailing list only includes recipients who have specifically signed up to receive your mail, and ensure that the email content itself is something your recipients actually want. Then, please email us with the details of your changes so that we can start the process of re-evaluating your case. If you are on probation and three weeks pass and we don't hear from you at all, we will have to disable your sending if we are still getting complaints about your mail. If you appealing a shutdown, then the information you send us needs to convince us that if you start sending again, the problem will no longer occur.

Amazon SES Spamtrap FAQ

Q1. What are spamtraps?

A spamtrap is a special email address maintained by an email provider, ISP, or anti-spam organization that is guaranteed not to have a human being behind it. Because that address will never legitimately be signed up to receive email, the organizations that maintain these spamtraps know that anyone who sends mail to any of these addresses is likely engaging in questionable email practices.

Q2. How are spamtraps set up?

Spamtrap addresses can be set up in multiple ways. They can be converted from addresses that were once valid, but have been unused (and bouncing) for an extended period of time. They can also be addresses that were set up just to be spamtraps. They can be unusual addresses that are hard to guess, and sometimes they are addresses that are close to real addresses (for example, introducing a typo into a common domain name). Often, but not always, spamtraps are "seeded" into the world by putting them on the internet in a variety of ways.

Q3. How does Amazon SES know if I am sending to spamtraps?

Certain organizations that operate spamtraps send Amazon SES notifications when their spamtraps are hit by Amazon SES senders.
Q4. How does Amazon SES use the spamtrap reports?

We review the reports, and if we find enough evidence that you have a problem with sending to spamtraps, we will put you on probation and ask you to fix the underlying problem. If you do not fix the problem in the probation period, your account will be suspended. Also, if your spamtrap problem is very severe, you might be immediately suspended without a probation period. As with any suspension, we will send you a notification at that time.

Q5. What should I do if I receive a probation or suspension notice for sending to spamtraps?

Fix the underlying problem and appeal to get your case reevaluated. For information about the appeal process, see the FAQs on probation and suspension. Due to the way spamtrap sending is reported, it will take a minimum of three weeks before we can confirm that a fix you have put in place has succeeded.

Q6. How many spamtrap hits can I have before I am put on probation or suspended?

Spamtrap hits are a very negative sign, so it takes only a small number of them to indicate that you are engaging in questionable sending practices.

Q7. Do you disclose the spamtrap addresses?

No. Spamtrap organizations disclose only the occurrence of spamtrap hits, not the actual spamtrap addresses. This is one of the measures they take to keep spamtrap addresses confidential and effective.

Q8. What can I do to avoid sending to spamtraps?

To reduce the risk of sending to spamtraps, follow these guidelines:

- Do not buy, rent, or share email addresses. Use only addresses that specifically requested your mail.
- Ensure that you ask for the email address twice to reduce the chance of typos.
- Use double opt-in to sign up new users. That is, when users sign up, send them a confirmation email that they need to click before receiving any additional mail.
- Ensure that you remove addresses that hard bounce from your list, so that they are removed long before they are converted to spamtraps.
- Ensure that you are monitoring engagement by your recipients, and stop sending to recipients who have not engaged with your emails or website recently. Time frames for what an "engaged user" is depend on your use case, but generally speaking if users haven't opened or clicked your emails in several months, you should consider removing them unless you have evidence that they do want your mail.
- Be very careful with re-engagement campaigns where you intentionally contact people who have not interacted with you recently. These efforts tend to be highly risky, and can often cause problems not only with spamtrap sending, but also with bounces and complaints.
- Send an opt-in message to your entire mailing list and keep only the recipients who click on the verification link. In addition to removing inactive recipients from your list, this procedure will remove spamtrap addresses as well. However, we do not recommend using this technique if you think that your mailing list might contain a lot of bad addresses and/or you already have a problem with bounces, because it might cause your bounce rate to reach the point at which your sending is put on probation or shut down.
Amazon SES Manual Investigation FAQ

Q1. I received a probation or shutdown notice for a manual investigation. What does that mean?

An Amazon SES investigator has identified a significant problem with your sending. Typical problems include, but are not limited to, the following:

• Your sending violates the AWS Acceptable Use Policy (AUP).
• Your emails appear to be unsolicited.
• Your content is associated with a use case that Amazon SES does not support.

If the problem is correctable, your account is put on probation and you are given a certain amount of time (rather than a certain volume of mail, as with bounces and complaints) to correct the problem. If the problem is uncorrectable, your account is suspended without a probation period.

Q2. Why would you do a manual investigation?

There are a variety of reasons. These include, but are not limited to, the following:

• Recipients contact Amazon SES to complain about your emails.
• We detect a significant change in your sending patterns.
• The spam filters of Amazon SES flag a significant portion of your emails.

The probation or suspension notification indicates the issue at a high level. For some problems, we are able to provide more specific details.

Q3. What are "unsolicited" emails?

Unsolicited emails are emails that the recipient did not specifically sign up for. This includes cases in which a recipient signs up for a certain type of mail (for example, notifications), and instead is sent a different type of mail (for example, advertisements). If the probation or suspension notice indicates that unsolicited sending is your problem, you should provide the following information in your appeal:

• Are all the messages that you send specifically requested by the recipient, and do they comply with the AUP?
• Have you acquired email addresses in any way other than a customer specifically interacting with you or your website and requesting emails from it? You should explain how you accumulated your mailing list.
• How do your subscribe and unsubscribe processes work? You should include your opt-in and opt-out links.

Also, feel free to provide any other information you might have to assure us that your email list was accumulated and is managed using the practices described in the Amazon Simple Email Service Email Sending Best Practices whitepaper.

Q4. What should I do if I receive a probation or suspension notice for a manual investigation?

As with any probation or suspension, fix the underlying problem that is causing the issue specified in the probation or suspension notice, and then appeal to get your case reevaluated. For information about the appeal process, see the FAQs on probation and suspension.
Q5. What types of problems do you view as "correctable?"

Generally, we believe the situation is correctable if you have a history of good sending practices, and if there are steps you can take to eliminate the problematic sending while continuing the bulk of your sending. For example, if you are sending three different types of email and only one type is problematic, you might be able to simply stop the problematic sending and continue with the rest of your sending.

Q6. What if I cannot find the source of the problem?

You can respond to the notification (or email ses-enforcement@amazon.com from the email address associated with your AWS account) and request a sample of the mail that caused the issue.

Amazon SES IP Blacklist FAQs

Mailbox providers aim to protect their users from receiving spam. To determine whether an incoming email is spam, mailbox providers examine various characteristics of the email. One of the methods they use is to check whether the IP address that sent the email is on an IP blacklist. An IP blacklist, also known as realtime blackhole list (RBL) or DNS-based blackhole list (DNSBL), is a list of IP addresses suspected to send spam. If the IP address from which the email was sent is on a blacklist that a mailbox provider uses, the mailbox provider might bounce the email or drop it entirely. Various organizations compile blacklists. Mailbox providers choose which of these blacklists they use. Some blacklists are more widely used than others.

As with IP addresses of other email service providers, Amazon SES IP addresses occasionally appear on blacklists. This topic describes how blacklists impact your sending with Amazon SES.

**Note**

This topic is about blacklists that mailbox providers use to block incoming mail from email service providers such as Amazon SES. If you are looking for information about how Amazon SES blocks outgoing mail to recipient addresses that have previously bounced, see Removing an Email Address from the Amazon SES Suppression List (p. 428).

Q1. How do I know if blacklisting is preventing recipients from receiving my emails?

If your sending is impacted by the blacklisting of an Amazon SES IP address, you will most likely receive a bounce notification (p. 226) that contains a message indicating that your email was rejected because of a listing on a blacklist. In most cases, the bounce message includes the name or URL of the blacklist. An example of this type of message is "Message rejected due to IP [0.0.0.0] listed on RBL [X]". If you do not see this type of message in your bounce notifications, it is unlikely that a blacklist is impacting your sending.

Q2. What does Amazon SES do about blacklisting?

We carefully and continually monitor the reputation of our IP address space. We work closely with mailbox providers and blacklist operators to identify and mitigate blacklistings in parallel with our routine compliance processes to remove offending users from our system.

Blacklist operators set their own listing and delisting policies, which do not always comply with the recommendations in RFC 6471, so we cannot provide a guarantee or timeframe in which a blacklisting will be resolved. In particular, we are aware that portions of our IP address space are sometimes listed on SORBS and UCEPROTECT. We frequently work with blacklist data to mitigate and resolve the listings, but we cannot estimate the time to resolution.
There are some blacklists, such as SORBS and SpamCannibal, that we are not currently working with to resolve listings. Our data indicates that the impact of these blacklists is extremely low. We are not aware of any major mailbox provider using SORBS or SpamCannibal to reject messages. If you are particularly concerned about SORBS, we recommend doing an internet search of “SORBS reliability” to get an idea of why ISPs should never use this blacklist as the sole criteria for rejecting mail.

Q3. Are all blacklists equally important?

No. Large mailbox providers typically use reputable blacklists such as Spamhaus, whereas smaller mailbox providers who aren’t aware of which blacklists are reputable might use less reputable blacklists. A Spamhaus listing is likely to result in mail being blocked at major mailbox providers, while a listing on SORBS,UCEPROTECT, or SpamCannibal might not have much impact on your sending. A service that asks you to pay for delisting or does not accept delist requests is unlikely to be used by major mailbox providers.

Q4. What should I do if I notice that an Amazon SES IP address is on a blacklist?

Rest assured that we are aware of the listing, and are working towards a resolution. However, remember that if you do not receive bounce notifications that contain a message about blacklisting, it is very unlikely that a blacklisting is impacting your sending. If your bounce notifications do indicate that a SORBS,UCEPROTECT, or SpamCannibal listing is impacting your sending, we recommend that you contact the mailbox provider and make sure that it is not using any of these as the sole criteria for blocking IP addresses.

Q5. Emails I send are being put in the junk folder. Could this be because an Amazon SES IP address is on a blacklist?

It is unlikely that your sending is impacted by blacklisting unless your bounce notifications contain a message indicating that blacklisting is the reason for the rejection of the message.

Amazon SES Email Sending Metrics FAQs

Amazon SES collects several metrics about the emails you send. These metrics enable you to analyze the effectiveness of your email program and monitor important statistics, such as your bounce and complaint rates.

This section contains FAQs on the following topics related to email sending metrics:

- the section called “General” (p. 451)
- the section called “Open Tracking” (p. 451)
- the section called “Click Tracking” (p. 452)
General Questions

Q1. After an email is delivered, how long will Amazon SES continue to collect open and click metrics?

Amazon SES collects open and click metrics for 60 days after each email is sent.

Q2. If a user opens an email multiple times, or clicks a link in an email multiple times, is each of those events tracked separately?

If a recipient opens an email multiple times, each instance is counted as a unique open event. Similarly, if a recipient clicks the same link multiple times, each click is counted as a unique click event.

Q3. Are open and click metrics aggregated, or can they be measured down to the recipient level?

Opens and clicks are tracked at the recipient level. With open and click tracking, you can determine which recipients opened an email or clicked a link in an email.

Q4. Can I retrieve open and click metrics using the Amazon SES API?

The Amazon SES API does not provide a method for retrieving open and click metrics. However, you can retrieve open and click metrics for Amazon SES using the CloudWatch API. For example, you can use the AWS CLI to retrieve click metrics using the CloudWatch API by issuing the following command:

```
aws cloudwatch get-metric-statistics --namespace AWS/SES --metric-name Click --statistics Sum --period 86400 --start-time 2017-01-01T00:00:00Z --end-time 2017-12-31T23:59:59Z
```

The command shown above retrieves the total number of click events for each day in 2017. To retrieve open metrics change the value of the `metric-name` parameter to `Open`. You can also modify the `start-time` and `end-time` parameters to change the analysis period, or change the `period` parameter for more fine-grained analysis.

Open Tracking

Q1. How does open tracking work?

At the bottom of each email sent through Amazon SES, we insert a 1 pixel by 1 pixel transparent GIF image. Each email includes a unique reference to this image file; when the image is opened, Amazon SES can tell exactly which message was opened and by whom.

The addition of this tracking pixel does not change the appearance of your email.

Q2. Is open tracking enabled by default?

Open tracking is available to all Amazon SES users by default. To use open tracking, you must do the following:

1. Create a configuration set.
2. In the configuration set, create an event destination.
3. Configure the event destination to publish open event notifications to a destination.
4. In every email for which you want to track opens, specify the configuration set that you created in step 1.

For a more detailed explanation of this process, see the section called “Monitoring Using Event Publishing” (p. 247).

Q3. Can I omit the open tracking pixel from certain emails?

There are two ways to omit the open tracking pixel from your emails. The first method is to send the email without specifying a configuration set. Alternatively, you can specify a configuration set that is not configured to publish data about open events.

Q4. Do you track opens for plaintext emails?

Open tracking only works with HTML emails. Because open tracking relies on the inclusion of an image, it is not possible to collect open metrics for users who open emails using a text-only (non-HTML) email client.

Click Tracking

Q1. How does click tracking work?

To track clicks, Amazon SES modifies each link in the body of the email. When recipients click a link, they are sent to an Amazon SES server, and are immediately forwarded to the destination address. As with open tracking, each redirect link is unique. This enables Amazon SES to determine which recipient clicked the link, when they clicked it, and the email from which they arrived at the link.

Important
If you send a single message to multiple recipients, each recipient will save the same click tracking link. To track individual recipients' click activity, send email to one recipient per send operation.

Q2. Can I disable click tracking?

You can disable click tracking by adding an attribute, `ses:no-track`, to the anchor tags in the HTML body of your email. For example, if you link to the AWS home page, a normal anchor link resembles the following:

```
<a href="https://aws.amazon.com">Amazon Web Services</a>
```

To disable click tracking for that same link, modify the link to resemble the following:

```
<a ses:no-track href="aws.amazon.com">Amazon Web Services</a>
```

Because `ses:no-track` is not a standard HTML attribute, we automatically remove it from the version of the email that arrives in your recipients' inboxes.

Q3. Is there a limit to the number of links that can be tracked in each email?

There is a limit to the number of links that can be tracked in a single email, although you are highly unlikely to encounter this limit in practice. Currently, the click tracking system will track a maximum of 250 links.
Q4. Are click metrics collected for links in plaintext emails?

In order to use click tracking, you must send HTML emails. Links in plaintext emails are not tracked by Amazon SES.

Q5. Can I tag links with unique identifiers?

You can add an unlimited number of tags, as key-value pairs, to links in your email by using the `ses:tags` attribute. When you use this attribute, specify the keys and values using the same format that you would use to pass inline CSS properties: type the key, followed by a colon (:) and a space, followed by the value. If you need to pass several key-value pairs, separate each pair with a semicolon (;).

For example, assume you want to add the tags `product:book`, `genre:fiction`, `subgenre:scifi`, `type:newrelease` to a link. The resulting link resembles the following:

```
<a ses:tags="product:book;genre:fiction;subgenre:scifi;type:newrelease;"
   href="http://www.amazon.com/…/">New Releases in Science Fiction</a>
```

These tags are passed through to your event publishing destination so that you can perform additional analysis on the specific links that your users clicked.

Q6. Do tracked links use the HTTP or HTTPS protocol?

Tracking links use the same protocol as the original links in your email.

For example, if your email includes a link to `https://www.amazon.com`, the link is replaced with a tracking link that uses the HTTPS protocol. If your email includes a link to `http://www.example.com`, the link is replaced with a tracking link that uses HTTP. If your email includes both of the previously mentioned links, the HTTPS link is replaced with a tracking link that uses the HTTPS protocol, and the HTTP link is replaced with a tracking link that uses the HTTP protocol.
# Amazon SES Resources

The following table lists resources that you may find useful as you work with Amazon Simple Email Service (Amazon SES).

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Simple Email Service API Reference</td>
<td>The Amazon SES API Reference. Contains complete descriptions of the API actions, parameters, and data types, and a list of errors that the service returns.</td>
</tr>
<tr>
<td>Amazon SES Pricing</td>
<td>Pricing information for Amazon SES.</td>
</tr>
<tr>
<td>SES Sending Limits Increase case</td>
<td>The Support Center form to request an increase in your sending limits and move out of the sandbox.</td>
</tr>
<tr>
<td>Request to Remove Email Sending Limitations</td>
<td>The form to request to remove the default Amazon EC2 sending limits.</td>
</tr>
<tr>
<td>Amazon SES Forum</td>
<td>The forum in which Amazon SES users can post questions and discuss various Amazon SES topics.</td>
</tr>
<tr>
<td>Amazon SES Blog</td>
<td>The blog that contains blog posts and announcements by the Amazon SES team.</td>
</tr>
<tr>
<td>AWS Developer Tools</td>
<td>Links to developer tools and resources that provide documentation, code samples, release notes, and other information to help you build innovative applications with AWS.</td>
</tr>
<tr>
<td>AWS Support Center</td>
<td>The hub for creating and managing your AWS Support cases. Also includes links to other helpful resources, such as forums, technical FAQs, service health status, and AWS Trusted Advisor.</td>
</tr>
<tr>
<td>Contact Us</td>
<td>A central contact point for inquiries concerning AWS billing, account, events, abuse, and other issues.</td>
</tr>
<tr>
<td>AWS Glossary</td>
<td>The AWS Glossary. Contains definitions of common terms used in Amazon SES and other AWS services.</td>
</tr>
<tr>
<td>Conditions of Use</td>
<td>AWS Acceptable Use Policy. Describes email abuse and other prohibited uses of the web services offered by Amazon Web Services, Inc.</td>
</tr>
</tbody>
</table>
| email-abuse@amazon.com                        | An email address for reporting malicious or unsolicited (spam) email sent from Amazon SES.  

When you contact this address, please provide the following information:

- The full headers of the email message. For more information about retrieving email headers, see https://support.google.com/mail/answer/22454?hl=en.
- The type of abuse you are experiencing. For example, unsolicited emails that do not provide a method of opting out.
Amazon SES Developer Guide
Appendix

This appendix contains supplementary information about sending emails through Amazon Simple Email Service (Amazon SES).

- For the header field requirements for emails that you send through Amazon SES, see Appendix: Header Fields (p. 455).
- For a list of attachment types that Amazon SES does not accept, see Appendix: Unsupported Attachment Types (p. 456).
- To learn how to create a basic bounce and complaint tracking dashboard using other AWS products, see Appendix: Create a Deliverability Dashboard (p. 457).

Appendix: Header Fields

Amazon SES accepts any email headers that follow the format described in RFC 822.

The following fields cannot appear more than once in a header:

- Accept-Language
- acceptLanguage (Note: This field is nonstandard. If possible, use Accept-Language instead.)
- Archived-At
- Auto-Submitted
- Bounces-to
- Comments
- Content-Alternative
- Content-Base
- Content-Class
- Content-Description
- Content-Disposition
- Content-Duration
- Content-ID
- Content-Language
- Content-Length
- Content-Location
- Content-MD5
- Content-Transfer-Encoding
- Content-Type
- Date (Note: Amazon SES overrides any Date header you provide with the time that Amazon SES accepts the message. The time zone of the Date header is UTC.)
- Delivered-To
- Disposition-Notification-Options
Appendix: Unsupported Attachment Types

You can send messages with attachments through Amazon SES by using the Multipurpose Internet Mail Extensions (MIME) standard. Amazon SES accepts all file attachment types except for attachments with the file extensions in the following list.
Note
Some ISPs have further limitations (such as restrictions regarding archived attachments), so we recommend testing your email sending through major ISPs before you send your production email.

Appendix: Create a Deliverability Dashboard

Bounce and complaint rates can have a negative impact on your sender reputation. A negative sender reputation makes it less likely that the messages you send using Amazon SES reach your recipients' inboxes. Additionally, if your bounce or complaint rate is too high, we may throttle or suspend your account to protect other users. For these reasons, it is essential that you have a process to remove email addresses that bounced or complained from your recipient list.

This section includes procedures for creating a simple dashboard that shows information about bounces and complaints. You can customize this solution to meet your specific needs. When you complete the procedures in this appendix, you will receive an email every day that includes a link to a web page stored in an Amazon S3 bucket. When you click the link, you see a dashboard that displays information about the numbers of bounces and complaints that were received in the previous 24-hour period. The dashboard also provides details about the messages that resulted in bounces and complaints. The image below shows an example of this dashboard.
There are several steps required to create and deploy this dashboard:

- **Part 1: Create a Topic in Amazon Simple Notification Service (p. 459)** contains procedures for creating an Amazon SNS topic. Amazon SNS topics are access points for people who want to receive specific notifications.

- **Part 2: Create a Queue in Amazon Simple Queue Service (p. 459)** contains procedures for creating an Amazon SQS queue. This Amazon SQS queue collects the bounce and complaint notifications from Amazon SNS and passes them to an AWS Lambda function for further processing.

- **Part 3: Create an Amazon Simple Storage Service Bucket (p. 460)** contains procedures for creating an Amazon Simple Storage Service (S3) bucket that stores the dashboards generated by the Lambda function.

- **Part 4: Create AWS Identity and Access Management Policies and Roles (p. 460)** contains procedures for setting up policies and roles in AWS Identity and Access Management (IAM). These policies and roles allow the components of this solution to work together securely.

- **Part 5: Configure Bounce and Complaint Notifications in Amazon SES (p. 462)** contains procedures for configuring Amazon SES to send bounce and complaint notifications to certain email addresses.

- **Part 6: Create an AWS Lambda Function (p. 462)** contains procedures for creating an AWS Lambda function. The function processes the bounce and complaint notifications in the Amazon SQS queue, creates new dashboards, and emails those dashboards to addresses you specify.

- **Part 7: Test the AWS Lambda Function (p. 464)** contains information about using the Amazon SES mailbox simulator to send sample bounce and complaint emails. After you send these emails, you can test the Lambda function to ensure it is working properly.

- **Part 8: Configure Triggers in Amazon CloudWatch (p. 464)** contains information about scheduling the Lambda function to run on a schedule that you specify.

As an alternative to completing the preceding sections, you can instead download an AWS CloudFormation template from the aws-support-tools repository on GitHub. To learn more about deploying AWS CloudFormation stacks, see Get Started in the AWS CloudFormation User Guide.
Part 1: Create a Topic in Amazon Simple Notification Service

The first step in creating a bounce and complaint tracking dashboard is to create a new topic in Amazon Simple Notification Service. When you create an Amazon SQS queue in the next section, you will subscribe to this topic to gather bounce and complaint notifications.

To create a new Amazon SNS topic
1. Sign in to the AWS Management Console and open the Amazon SNS console at https://console.aws.amazon.com/sns/v2/home.
2. In the navigation bar, choose Topics.
3. Choose Create new topic.
4. For Topic name, enter a name for the topic.
5. For Display name, enter a display name for the topic.
6. Choose Create topic.
7. Proceed to Part 2: Create a Queue in Amazon Simple Queue Service (p. 459).

Part 2: Create a Queue in Amazon Simple Queue Service

In this procedure, you create a new queue in Amazon SQS and subscribe to the topic you created in the previous procedure. This queue gathers all of the bounce and complaint notifications you receive; an AWS Lambda then collects the notifications in this queue for further processing.

To create a new Amazon SQS queue
1. Open the Amazon SQS console at https://console.aws.amazon.com/sqs/.
2. If you have not created an Amazon SQS queue in the past, choose Get Started Now.
   Otherwise, if your AWS account already contains one or more Amazon SQS queues, choose Create New Queue.
3. Under What do you want to name your queue?, for Queue Name, type a name for the queue.
5. Choose Configure Queue.
6. For Default Visibility Timeout, choose 5 minutes.
7. Leave the remaining attributes at their default settings. Choose Create Queue.
8. In the list of queues, select the queue you just created. On the Queue Actions menu, choose Subscribe Queue to SNS Topic.
9. On the Subscribe to a Topic window, make the following selections:
   - For Topic Region, choose the AWS Region that contains the Amazon SNS topic you created in Part 1: Create a Topic in Amazon Simple Notification Service (p. 459).
   - For Choose a Topic, choose the topic you created in Part 1: Create a Topic in Amazon Simple Notification Service (p. 459).
10. Choose Subscribe.
Part 3: Create an Amazon Simple Storage Service Bucket

This procedure describes how to create and configure an Amazon S3 bucket to store the dashboard files that are generated by the AWS Lambda function.

To create a new Amazon S3 bucket
1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Choose Create bucket.
3. On the Create bucket window, for Bucket name, type a name for the bucket.
4. For Region, choose the AWS Region that contains your Amazon SNS topic and Amazon SQS queue. Choose Create.

Part 4: Create AWS Identity and Access Management Policies and Roles

To ensure the security of your AWS account, you must create an AWS Identity and Access Management (IAM) policy and role. The policy and role define the ways that the components of this solution can interact with each other. This procedure describes how to configure these policies and roles.

To create a new IAM policy and role
1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation bar, choose Policies.
3. Choose Create policy.
4. On the JSON tab, paste the following code into the editor:

```json
{
"Version": "2012-10-17",
"Statement": [
{
 "Sid": "AllowSendEmail",
 "Effect": "Allow",
 "Action": [
 "ses:SendEmail"
 ],
 "Resource": ["*"]
 },
{
 "Sid": "s3allow",
 "Effect": "Allow",
 "Action": [
 "s3:PutObject",
 "s3:PutObjectAcl"
 ],
 "Resource": [
 "arn:aws:s3:::BUCKET_NAME/*"
 ]
 },
{
 "Sid": "AllowQueuePermissions",
 "Effect": "Allow",
 "Action": [
 "sqs:SendMessage",
 "sqs:ReceiveMessage"
 ],
 "Resource": ["*"
 ]
 }
}
```
In the pasted code, change the following attributes:

- Replace `BUCKET_NAME` with the name of the Amazon S3 bucket you created in Part 3: Create an Amazon Simple Storage Service Bucket (p. 460).
- Replace `SQS_QUEUE_ARN` with the ARN of the Amazon SQS queue you created in Part 2: Create a Queue in Amazon Simple Queue Service (p. 459).

**Tip**
You can find the ARN of the Amazon SQS queue by completing the following steps:

1. In the Amazon SQS console, choose the queue you created in Part 2: Create a Queue in Amazon Simple Queue Service (p. 459).
2. In the Details section, copy the value shown next to ARN. The format of the ARN should resemble the following example:

   `arn:aws:sqs:us-east-1:999623213###:sample-queue-name`

5. Choose Review policy.
6. On the Review policy screen, complete the following sections:
   - For Name, type a name for the policy.
   - For Description, type a brief description of the policy.
7. Choose Create policy.
8. In the navigation bar, choose Roles.
9. Choose Create role.
10. Under Select type of trusted entity, choose Lambda, and then choose Next: Permissions.
11. On the Attach permissions policies screen, check the box next to the name of the policy you just created, and then choose Next: Review.
12. On the Review screen, for Role name, type a name for the role, and then choose Create role.
13. Proceed to Part 5: Configure Bounce and Complaint Notifications in Amazon SES (p. 462).
Part 5: Configure Bounce and Complaint Notifications in Amazon SES

To receive bounce and complaint notifications from Amazon SES, you must configure notifications in the Amazon SES console. This procedure describes how to set up these notifications.

1. Open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the navigation bar, under Identity Management, choose Email Addresses.
3. Select the email address that you want to use to receive bounce and complaint notifications, and then choose View Details.
5. For both Bounces and Complaints, choose the Amazon SNS topic you created in Part 1: Create a Topic in Amazon Simple Notification Service (p. 459). Choose Save Config.
6. Repeat steps 3–5 for each email address that should receive bounce or complaint notifications.
7. Open the Amazon SQS console at https://console.aws.amazon.com/sqs/.
8. Right-click the queue you created in Part 2: Create a Queue in Amazon Simple Queue Service (p. 459), and then choose Purge Queue.

Note
This step is required. When you configure notifications, Amazon SES sends a confirmation notification that is picked up by Amazon SNS. These confirmations cannot be parsed by the Lambda function, and may cause the Lambda function to fail.


Part 6: Create an AWS Lambda Function

Now that all of the components are in place, you can create a Lambda function. The function creates an HTML dashboard and notifies you by email when that dashboard is updated.

To create a new AWS Lambda function

2. Open the AWS Lambda console at https://console.aws.amazon.com/lambda/.
3. In the navigation bar, choose Functions.
4. Choose Create function.
5. Choose Author from scratch.
6. Under Author from scratch page, complete the following steps:
   - For Name, type a name for the function.
   - For Runtime, choose Node.js 6.10.
   - For Role, select Choose an existing role.
   - For Existing role, choose the IAM role you created in Part 4: Create AWS Identity and Access Management Policies and Roles (p. 460).

   Choose Create function.
9. Under Environment variables, add the following keys and values:
Keys and Values

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>Sample Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueueURL</td>
<td>The URL of the Amazon SQS queue you created in Part 2: Create a Queue in Amazon Simple Queue Service (p. 459).</td>
<td><a href="https://sqs.us-east-1.amazonaws.com/999623213###/sample-queue-name">https://sqs.us-east-1.amazonaws.com/999623213###/sample-queue-name</a></td>
</tr>
<tr>
<td>Region</td>
<td>The AWS Region in which you created the Amazon SNS topic in Part 1: Create a Topic in Amazon Simple Notification Service (p. 459).</td>
<td>us-east-1</td>
</tr>
<tr>
<td>ToAddr</td>
<td>[Optional] The email address that will receive the bounce and complaint report.</td>
<td><a href="mailto:robert@example.com">robert@example.com</a></td>
</tr>
<tr>
<td>SrcAddr</td>
<td>[Optional] The email address that will send the bounce and complaint report.</td>
<td><a href="mailto:reports@example.com">reports@example.com</a></td>
</tr>
<tr>
<td>BucketName</td>
<td>The name of the Amazon S3 bucket you created in Part 3: Create an Amazon Simple Storage Service Bucket (p. 460).</td>
<td>sample-s3-bucket</td>
</tr>
<tr>
<td>BucketPrefix</td>
<td>[Optional] If you want to save the dashboards in a folder in the Amazon S3 bucket, specify the path here. The path you specify must end with a forward slash (/).</td>
<td>SES/reports/</td>
</tr>
</tbody>
</table>

10. Under **Basic settings**, make the following selections:

   • For **Memory (MB)**, choose **512 MB**.

   **Note**
   If you think you will receive more than 5,000 bounces and complaints per day, increase the value in the **Memory** field to at least 1024 MB. If you think you will receive more than 10,000 bounces and complaints per day, increase the value in the **Memory** field to at least 1536 MB.

   • For **Timeout**, select **5 min**.

11. Choose **Test**.

12. On the **Configure test event** window, for **Event name**, type a name, and then choose **Create**.

13. In the notification bar at the top of the page, confirm that the function ran successfully. If it did not, ensure that you entered the correct values for the environment variables in step 9.

14. Proceed to **Part 7: Test the AWS Lambda Function (p. 464)**.
Part 7: Test the AWS Lambda Function

Before you schedule the Lambda function to run on a regular basis, you should test it to be sure that all of the components are configured properly. In this procedure, you send two messages through the Amazon SES mailbox simulator, and then run the Lambda function to ensure that messages are being processed properly.

To test the Lambda function

1. Open the Amazon SES console at https://console.aws.amazon.com/ses/.
2. In the column on the left, choose either Email Addresses or Domains.
3. Select a verified email address or domain, and then choose Send a Test Email.
4. For To, type bounce@simulator.amazonses.com. For Subject and Body, type some sample text. Choose Send Test Email.
5. Repeat steps 3 and 4 to create another test message, but this time, for To, type complaint@simulator.amazonses.com.
6. Open the Amazon SQS console at https://console.aws.amazon.com/sqs/. The Messages Available column should indicate that 2 messages are available.
8. In the navigation bar, choose Functions.
9. In the list of functions, choose the function you created in Part 6: Create an AWS Lambda Function (p. 462).
10. Choose Test. When the function finishes running, expand the Details section. If the Lambda function was configured properly, you will receive one of the following messages:
   - null: Indicates that the function ran without errors.
   - Queue empty: Indicates that there were no new bounce or complaint notifications in the queue.
11. Proceed to Part 8: Configure Triggers in Amazon CloudWatch (p. 464).

Part 8: Configure Triggers in Amazon CloudWatch

After you validate that the Lambda function is working properly, you can use Amazon CloudWatch to schedule the function to run every day. Each time the Lambda function runs, you receive an email that includes a link to a web page stored in an Amazon S3 bucket. When you click the link, you see a dashboard that displays information about the numbers of bounces and complaints that were received in the previous 24-hour period.

To configure trigger conditions in Amazon CloudWatch

2. In the navigation bar, under Events, choose Rules.
3. Choose Create rule.
5. For Fixed rate of, choose 1 Day.
7. For Function, choose the Lambda function you created in Part 6: Create an AWS Lambda Function (p. 462), and then choose Configure details.
8. Under Rule definition, for Name, type a name for the rule, and then choose Create rule.
# Amazon SES Developer Guide

## Document History

The following table lists the major changes to the *Amazon Simple Email Service (Amazon SES) Developer Guide*.

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date Changed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation update</td>
<td>Revised the Amazon SNS event publishing field definitions (p. 273), and added a Rendering Failure event example (p. 288).</td>
<td>January 22, 2018</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Updated Deliverability Dashboard appendix (p. 457) to account for changes to IAM and Lambda consoles.</td>
<td>January 18, 2018</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Updated content related to publishing events to CloudWatch (p. 249) to mention blacklisted fields.</td>
<td>January 15, 2018</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Updated procedures for sending email using OpenSSL (p. 95) to make them easier to follow.</td>
<td>January 11, 2018</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Added code example for sending raw email by using the AWS SDK for Ruby.</td>
<td>January 2, 2018</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Added code example for sending raw email by using the AWS SDK for PHP.</td>
<td>December 29, 2017</td>
</tr>
<tr>
<td>New feature</td>
<td>Added content related to custom verification emails.</td>
<td>December 7, 2017</td>
</tr>
<tr>
<td>New feature</td>
<td>Added content related to pausing email sending and exporting reputation metrics for configuration sets.</td>
<td>November 15, 2017</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Added code example for sending raw email by using the AWS SDK for Java.</td>
<td>October 23, 2017</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Added code example for sending raw email by using the AWS SDK for Python (Boto).</td>
<td>October 20, 2017</td>
</tr>
<tr>
<td>New feature</td>
<td>Added content related to the email templates and personalized email features.</td>
<td>October 11, 2017</td>
</tr>
<tr>
<td>New feature</td>
<td>Added content related to the open and click custom domain feature.</td>
<td>September 18, 2017</td>
</tr>
<tr>
<td>New feature</td>
<td>Added content related to the reputation dashboard.</td>
<td>August 24, 2017</td>
</tr>
<tr>
<td>New feature</td>
<td>Added content related to dedicated IP pools feature.</td>
<td>August 17, 2017</td>
</tr>
<tr>
<td>New feature</td>
<td>Added content related to open and click tracking feature.</td>
<td>August 1, 2017</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Added an index of code examples.</td>
<td>June 26, 2017</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date Changed</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Added an appendix that demonstrates the process of creating a deliverability dashboard for Amazon SES.</td>
<td>June 22, 2017</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Updated email sending code examples.</td>
<td>June 6, 2017</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for dedicated IPs.</td>
<td>November 21, 2016</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for email sending event publishing.</td>
<td>November 2, 2016</td>
</tr>
<tr>
<td>Service update</td>
<td>Updated to reflect that users no longer need to explicitly enable Easy DKIM signing after generating their DKIM records.</td>
<td>September 15, 2016</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Added a getting started tutorial for receiving email.</td>
<td>July 12, 2016</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for enhanced notifications.</td>
<td>June 14, 2016</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for custom MAIL FROM domains.</td>
<td>March 14, 2016</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for inbound email.</td>
<td>September 28, 2015</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for sending authorization.</td>
<td>July 8, 2015</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for AWS CloudTrail logging.</td>
<td>May 7, 2015</td>
</tr>
<tr>
<td>Service update</td>
<td>Updated to reflect the consolidation of the Amazon SES limit increase forms and removed 'production access' terminology.</td>
<td>April 8, 2015</td>
</tr>
<tr>
<td>Service update</td>
<td>Updated with new requirements for domain verification TXT records.</td>
<td>February 25, 2015</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Added Enforcement FAQ.</td>
<td>December 15, 2014</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for delivery notifications.</td>
<td>June 23, 2014</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for subdomain support.</td>
<td>March 19, 2014</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for Amazon SES expansion to the US West (Oregon) region.</td>
<td>January 29, 2014</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for Amazon SES expansion to the EU (Ireland) region.</td>
<td>January 15, 2014</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated to reflect the changes in validation of Header Fields and MIME Types.</td>
<td>November 6, 2013</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Removed content on Sender ID.</td>
<td>August 22, 2013</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated to reflect the Amazon SES console redesign.</td>
<td>June 19, 2013</td>
</tr>
<tr>
<td>New feature</td>
<td>Replaced the blacklist with the suppression list.</td>
<td>May 8, 2013</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date Changed</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for the blacklist removal feature.</td>
<td>March 4, 2013</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Added MIME types.</td>
<td>February 4, 2013</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Included a Getting Started section to replace the stand-alone Getting Started guide, restructured the Table of Contents, and updated the Sendmail integration instructions.</td>
<td>January 21, 2013</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Added troubleshooting sections on increasing throughput and SMTP issues.</td>
<td>December 12, 2012</td>
</tr>
<tr>
<td>Documentation update</td>
<td>Restructured the information on sending limits.</td>
<td>November 9, 2012</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for the Amazon SES mailbox simulator.</td>
<td>October 3, 2012</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for using a DKIM signature to sign email from a verified identity.</td>
<td>July 17, 2012</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for receiving bounce and complaint feedback notifications through Amazon Simple Notification Service (Amazon SNS).</td>
<td>June 26, 2012</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for domain verification.</td>
<td>May 15, 2012</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated to reflect additional header and attachment types.</td>
<td>April 25, 2012</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for the STARTTLS extension to SMTP.</td>
<td>March 7, 2012</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for SMTP support.</td>
<td>December 13, 2011</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for AWS Management Console support.</td>
<td>November 17, 2011</td>
</tr>
<tr>
<td>New feature</td>
<td>Updated for attachment support.</td>
<td>July 18, 2011</td>
</tr>
<tr>
<td>Initial release</td>
<td>This is the first release of the Amazon Simple Email Service Developer Guide.</td>
<td>January 25, 2011</td>
</tr>
</tbody>
</table>